Journal of Artificial Intelligence Research 55 (2016) 653-683 Submitted 06/15; published 03/16

Exact Algorithms for MRE Inference

Xiaoyuan Zhu XIAOYUAN.ZHU@QC.CUNY.EDU
Changhe Yuan CHANGHE.YUAN@QC.CUNY.EDU
Queens College, City University of New York
65-30 Kissena Blvd., Queens, NY 11367

Abstract

Most Relevant Explanation (MRE) is an inference task in Bayesian networks that finds the most relevant partial instantiation of target variables as an explanation for given evidence by maximizing the Generalized Bayes Factor (GBF). No exact MRE algorithm has been developed previously except exhaustive search. This paper fills the void by introducing two Breadth-First Branch-and-Bound (BFBnB) algorithms for solving MRE based on novel upper bounds of GBF. One upper bound is created by decomposing the computation of GBF using a target blanket decomposition of evidence variables. The other upper bound improves the first bound in two ways. One is to split the target blankets that are too large by converting auxiliary nodes into pseudo-targets so as to scale to large problems. The other is to perform summations instead of maximizations on some of the target variables in each target blanket. Our empirical evaluations show that the proposed BFBnB algorithms make exact MRE inference tractable in Bayesian networks that could not be solved previously.

1. Introduction

Bayesian networks are probabilistic models that capture the conditional independencies between random variables as directed acyclic graphs, and provide principled approaches to scientific explanation. Explanation tasks in Bayesian networks can be classified into three categories: explanation of reasoning, explanation of model, and explanation of evidence (Lacave & Diez, 2002). The goal of *explanation of reasoning* in Bayesian networks is to explain the reasoning process used to produce the results so that the credibility of the results can be established. The goal of *explanation of model* is to present the knowledge encoded in a Bayesian network in easily understandable forms such as visual aids so that experts or users can examine or even update the knowledge. The goal of *explanation of evidence* is to explain why some observed variables are in their particular states using the other variables in the domain.

This research focuses on developing algorithms for solving one of the methods for explaining evidence in Bayesian networks, the Most Relevant Explanation (MRE) (Yuan & Lu, 2007; Yuan, Lim, & Lu, 2011b). The idea of MRE is to find a partial instantiation of the target variables that maximizes the Generalized Bayes Factor (GBF) (Fitelson, 2007; Good, 1985) as an explanation for the evidence. GBF is a rational function of probabilities that is suitable for comparing explanations with different cardinalities. MRE is shown both theoretically and empirically to be able to prune away independent and less relevant variables from the final explanation (Yuan et al., 2011b; Yuan, Liu, Lu, & Lim, 2009; Pacer, Lombrozo, Griffiths, Williams, & Chen, 2013).

Due to the difficulty of finding a meaningful upper bound for GBF, no exact algorithms have been developed to solve MRE except exhaustive search. Only local search and Markov chain Monte Carlo methods have been proposed previously (Yuan et al., 2009; Yuan, Lim, & Littman, 2011a). In this paper, we introduce the first non-trivial exact MRE algorithms based on Breadth-First Branch-and-Bound (BFBnB) search. The key idea of the proposed methods is to decompose the whole Bayesian network into a set of overlapping subnetworks using a *target blanket decomposition of evidence variables*. Each subnetwork is characterized by a subset of evidence variables and its target blanket which d-separates the evidence variables from other target and evidence variables. An upper bound for GBF is derived by solving independent optimization problems in the subnetworks. We also show that the bound can be tightened by merging the target blankets that share target variables.

Then we propose another improved upper bound based on two novel ideas. First, the above decomposition may lead to large target blankets that prevent the branch-and-bound algorithm from scaling to large MRE problems. To address this problem, we propose to split large target blankets by converting auxiliary nodes into *pseudo-targets* which introduce additional decomposition. Second, we find that the upper bound can be tightened by identifying and summing out *enclosed-targets* from each target blanket. The proposed upper bounds are used in the BFBnB algorithms for pruning the search space. We evaluated the algorithms in a set of benchmark diagnostic Bayesian networks. Experimental results show that the proposed algorithms make exact MRE inference tractable in Bayesian networks that could not be solved previously.

The rest of the paper is organized as follows. The basics of Bayesian networks and the MRE problem are introduced in Section 2. In Section 3, a novel target blanket upper bound is proposed. An improved upper bound is discussed in Section 4. The proposed BFBnB algorithms are introduced in Section 5. In Section 6, the experimental results are presented. Finally, discussions and conclusions are provided in Section 7.

2. Background

In this section, we introduce the basics of Bayesian networks, explanation in Bayesian networks, and the MRE problem.

2.1 Bayesian Networks and Moral Graph

A Bayesian network (Pearl, 1988; Darwiche, 2009; Koller & Friedman, 2009) is represented as a directed acyclic graph (DAG). The nodes in the DAG represent random variables. The lack of arcs in the DAG define conditional independence relations among the nodes. If there is an arc from node Y to X, we say that Y is a parent of X, and X is a child of Y. We use upper-case letters to denote variables X or variable sets $\mathbf{X}$, and lower-case letters for values of scalars x or vectors $\mathbf{x}$. A node Y is an ancestor of a node X if there is a directed path from Y to X. Let $AN(X)$ denote all the ancestors of X, then the smallest ancestral set $AN(\mathbf{X})$ of node set $\mathbf{X}$ is defined as $AN(\mathbf{X}) = \mathbf{X} \cup (\cup_{X_i \in \mathbf{X}} AN(X_i))$. In directed graphs, *d-separation* describes the conditional independence relation between two sets of nodes $\mathbf{X}$ and $\mathbf{Y}$, given a third set of nodes $\mathbf{Z}$, i.e., $p(\mathbf{X}|\mathbf{Z}, \mathbf{Y}) = p(\mathbf{X}|\mathbf{Z})$. The Markov blanket of $\mathbf{X}$ is the smallest node set which d-separates $\mathbf{X}$ from the remaining nodes in the network. The network as a whole represents the joint probability distribution of $\prod_X p(X|PA(X))$, where $PA(X)$ is the set of all the parents of X.

The moral graph G^m of a DAG G is an undirected graph with the same set of nodes. There is an edge between X and Y in G^m if and only if there is an edge between them in G or if they are parents of the same node in G. In an undirected graph, $\mathbf{Z}$ separates $\mathbf{X}$ and $\mathbf{Y}$, if $\mathbf{Z}$ intercepts all paths between $\mathbf{X}$ and $\mathbf{Y}$. Moral graphs are a powerful construction to explain d-separation.

Lemma 1 (Lauritzen, Dawid, Larsen, & Leimer., 1990) links d-separation in DAG to separation in undirected graphs.

Lemma 1. *Let **X**, **Y**, and **Z** be disjoint subsets of nodes in a DAG G. Then **Z** d-separates **X** from **Y** if and only if **Z** separates **X** from **Y** in* $(G_{AN(\boldsymbol{X}\cup\boldsymbol{Y}\cup\boldsymbol{Z})})^m$*, where* $(G_{AN(\boldsymbol{X}\cup\boldsymbol{Y}\cup\boldsymbol{Z})})^m$ *is the moral graph of the subgraph of G with node set* $AN(\boldsymbol{X} \cup \boldsymbol{Y} \cup \boldsymbol{Z})$.

2.2 Explanation of Evidence in Bayesian Networks

Numerous methods have been developed to explain evidence in Bayesian networks. Some of these methods make simplifying assumptions and focus on singleton explanations (Heckerman, Breese, & Rommelse, 1995; Jensen & Liang, 1994; Kalagnanam & Henrion, 1988). However, singleton explanations may be *underspecified* and are unable to fully explain the given evidence if the evidence is the compound effect of multiple causes.

For a domain with multiple interdependent target variables, multivariate explanations are often more appropriate for explaining the given evidence. *Maximum a Posteriori assignment* (MAP) finds a complete instantiation of a set of target variables that maximizes the joint posterior probability given partial evidence on the other variables. Recently Kwishthout (2013) extended MAP to find a set of joint assignments as *most inforbable explanations*. *Most Probable Explanation* (MPE) (Pearl, 1988) is similar to MAP except that MPE defines the target variables to be all unobserved variables. The common drawback of these methods is that they often produce hypotheses that are *overspecified* and may contain irrelevant variables in explaining the given evidence.

Everyday explanations are necessarily partial explanations (Leake, 1995). Various pruning techniques have been used to avoid overly complex explanations. These methods can be grouped into two categories: *pre-pruning* and *post-pruning*. *Pre-pruning* methods use the context-specific independence relations represented in Bayesian networks to prune irrelevant variables (Pearl, 1988; Shimony, 1993; van der Gaag & Wessels, 1993, 1995) before applying methods such as MAP to generate explanations. In contrast, *post-pruning* methods first generate explanations using methods such as MAP or MPE and then prune variables that are not important. An example is the method proposed by de Campos et al. (2001).

Several methods aim to directly find more appropriate explanations. The likelihood of evidence is used to measure the explanatory power of an explanation (Gärdenfors, 1988). Chajewska and Halpern (1997) extend the approach further to use the value pair of $<$likelihood, prior probability$>$ to order the explanations, forcing users to make decisions if there is no clear order between two explanations. Henrion and Druzdzel (1991) assume that a system has a set of pre-defined explanation scenarios organized as a tree; they use the scenario with the highest posterior probability as the explanation. Flores et al. (2005) propose to automatically create an explanation tree by greedily branching on the most informative variable at each step while maintaining the probability of each branch of the tree above a certain threshold. Nielsen et al. developed another method that uses the *causal information flow* (Ay & Polani, 2008) to select variables to expand an explanation tree.

2.3 Most Relevant Explanation

For explanation of evidence in Bayesian networks, we often classify the nodes into three categories: target, evidence, and auxiliary. The target set $\mathbf{M}$ represents variables of interest in the inference. The evidence set $\mathbf{E}$ represents observed information. The auxiliary set represents the variables that are not of interest in the inference. MRE, which finds a partial instantiation of $\mathbf{M}$ as an explanation for given evidence $\mathbf{e}$ in a Bayesian network, is formally defined as follows (Yuan et al., 2011b).

Definition 1. *Let* $\mathbf{M}$ *be a set of targets, and* $\mathbf{e}$ *be the given evidence in a Bayesian network.* Most Relevant Explanation *is the problem of finding a partial instantiation* $\mathbf{x}$ *of* $\mathbf{M}$ *that has the maximum generalized Bayes factor score* $GBF(\mathbf{x};\mathbf{e})$ *as the explanation for* $\mathbf{e}$*, i.e.,*

$$MRE(\mathbf{M};\mathbf{e}) = \underset{\mathbf{x},\emptyset\subset\mathbf{X}\subseteq\mathbf{M}}{\operatorname{argmax}}\ GBF(\mathbf{x};\mathbf{e}), \tag{1}$$

with GBF defined as

$$GBF(\mathbf{x};\mathbf{e}) = \frac{p(\mathbf{e}|\mathbf{x})}{p(\mathbf{e}|\bar{\mathbf{x}})}^{1}, \tag{2}$$

where $\mathbf{x}$ *is the joint value assignment (instantiation) of a subset* $\mathbf{X}$ *of* $\mathbf{M}$*, and* $\bar{\mathbf{x}}$ *represents all alternative explanations of* $\mathbf{x}$*.*

A commonly used measure for selecting explanatory hypothesis is the probability of an explanation given the evidence, as used in MAP and MPE to find the most likely configuration of a set of target variables. Probability-based methods, however, do not have the intrinsic capability to prune less relevant facts. Moreover, the probability measure is quite sensitive to the modeling choices, e.g., simply refining a model can dramatically change the best explanation. In contrast, MRE maximizes the rational function of probabilities GBF in Equation 2. This makes it possible to compare explanations with different cardinalities and to prune the less relevant variables automatically in a principled way. The search space of MRE is exponential in the number of targets, and the complexity of MRE is conjectured to NP^{PP}-hard (Yuan et al., 2011a), which makes the naive brute force algorithm impractical.

To further study the properties of generalized Bayes factor, we reformulate GBF as follows.

$$\begin{aligned} GBF(\mathbf{x};\mathbf{e}) &= \frac{p(\mathbf{e}|\mathbf{x})}{p(\mathbf{e}|\bar{\mathbf{x}})} = \frac{p(\mathbf{x}|\mathbf{e})p(\bar{\mathbf{x}})}{p(\mathbf{x})p(\bar{\mathbf{x}}|\mathbf{e})} \\ &= \frac{p(\mathbf{x}|\mathbf{e})(1-p(\mathbf{x}))}{p(\mathbf{x})(1-p(\mathbf{x}|\mathbf{e}))}. \end{aligned} \tag{3}$$

Therefore, we only need the prior and posterior probabilities of a hypothesis in order to compute its GBF. GBF is hence able to overcome the drawback of Bayes factor (Jeffreys, 1961) in having to do pairwise comparisons between multiple hypotheses.

Belief update ratio is a useful concept. The belief update ratio of $\mathbf{X}$ given $\mathbf{e}$ is defined as follows (Yuan et al., 2011b).

$$r(\mathbf{X};\mathbf{e}) = \frac{p(\mathbf{X}|\mathbf{e})}{p(\mathbf{X})}. \tag{4}$$

GBF can be calculated from the belief update ratio as follows.

$$\begin{aligned} GBF(\mathbf{x};\mathbf{e}) &= \frac{p(\mathbf{x}|\mathbf{e})(1-p(\mathbf{x}))}{p(\mathbf{x})(1-p(\mathbf{x}|\mathbf{e}))} = \frac{r(\mathbf{x};\mathbf{e})-p(\mathbf{x}|\mathbf{e})}{1-p(\mathbf{x}|\mathbf{e})} \\ &= 1+\frac{r(\mathbf{x};\mathbf{e})-1}{1-p(\mathbf{x}|\mathbf{e})}. \end{aligned} \tag{5}$$

1. We use $p(\mathbf{x})$ as a shorthand for $p(\mathbf{X}=\mathbf{x})$ in this paper.

2.4 Existing Methods for Solving MRE

Local search methods such as tabu search (Glover, 1990) have been applied to solve MRE (Yuan et al., 2011a). Tabu search starts with the empty solution set. At each step, it generates the neighbors of the current solution by adding, changing, or deleting one target variable. Then tabu search selects the best neighbor which has the highest GBF score and has not been visited before. In tabu search, the best neighbor can be worse than the current solution. To stop tabu search properly, upper bounds are set on both the total number of search steps M and the number of search steps since the last improvement L as the stopping criteria. Another Markov Chain Monte Carlo algorithm integrates the reversible-jump MCMC algorithm (Green, 1995) and simulated annealing (Kirkpatrick, Gelatt, & Vecchi, 1983) to find a solution by simulating a non-homogeneous Markov chain that eventually concentrates its mass on the mode of a distribution of the GBF scores of all solutions. These methods can only provide approximate solutions whose quality are unknown. Furthermore, the accuracy and efficiency of these methods are typically highly sensitive to tunable parameters.

2.5 Branch and Bound Algorithms for Solving MPE/MAP

Branch-and-bound algorithms have been developed for solving MAP and MPE by using upper bounds derived based on the property of optimization criterion or the structure of Bayesian networks. A mini-bucket upper bound was used in an AND/OR tree search for solving MPE (Dechter & Rish, 2003; Marinescu & Dechter, 2009). Recently, an improved mini-bucket upper bound (Marinescu, Dechter, & Ihler, 2014) was proposed to guide AND/OR search for exact MAP inference. The work in (Choi, Chavira, & Darwiche, 2007) showed that the mini-bucket upper bound can be derived from a node splitting scheme. To solve MAP exactly, an upper bound is proposed by commuting the order of max and sum operations in the MAP calculation (Park & Darwiche, 2003). An exact algorithm is proposed for solving MAP by computing upper bounds on an arithmetic circuit compiled from a Bayesian network (Huang, Chavira, & Darwiche, 2006).

3. A Novel Upper Bound Based on Target Blanket Decomposition for Solving MRE

It is difficult to solve MRE problems exactly because of both an exponential search space and the need for probabilistic inference at each search step. A naive brute-force search method can scale to Bayesian networks with at most 15 targets. In this work, we develop breadth-first branch-and-bound algorithms that use a suite of new upper bounds based on target blanket decomposition to prune the search space. The algorithm makes it possible to solve MRE problems with more targets exactly.

3.1 Search Space Formulation

Assuming there are n targets, and each target has d states, the search space of MRE contains $(d+1)^n - 1$ possible states (or solutions). We organize the search space as a search tree by instantiating the targets according to a total order π of the targets. Each state S in the tree contains values of a subset of the targets $\mathbf{V}$. $\mathbf{V}$ is defined as the *expanded* set of targets. The set of variables $\mathbf{U}$ that are yet to be considered for expansion is defined as *unexpanded* set; i.e., $\mathbf{U} = \{U_i | U_i \in \mathbf{M};\ \forall V_j \in \mathbf{V},\ V_j <_\pi U_i\}$; and the set of variables $\mathbf{P}$ that have been considered but are not used is defined as *pruned* set; i.e., $\mathbf{P} = \mathbf{M} \setminus \{\mathbf{V} \cup \mathbf{U}\}$. Figure 1 demonstrates the different target sets for the state $\{X_1, X_4, X_6\}$ for a 9-target MRE problem.

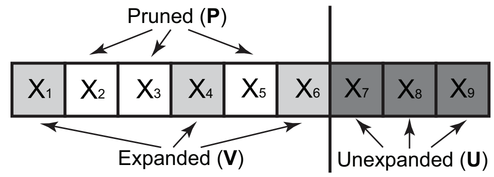

Figure 1: Different types of targets of a search state in MRE, i.e., expanded (gray), pruned (white), and unexpanded (black).

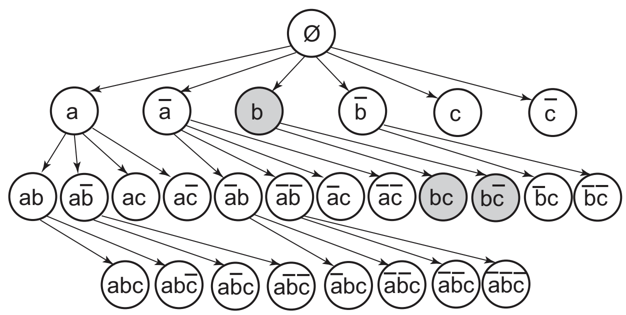

Figure 2: The search tree of an MRE problem with three targets, i.e., A, B, and C. An example of sub-tree rooted at $\{b\}$ is marked as gray.

The search tree has the empty state as the root. Each non-leaf state in the tree has a number of children states that instantiate one more unexpanded target. Figure 2 shows an example search tree with three targets $A = \{a, \bar{a}\}$, $B = \{b, \bar{b}\}$, and $C = \{c, \bar{c}\}$ with π in the same order. Different branches of the search tree may have different numbers of layers, but all of the states with the same cardinality appear in the same layer.

It is possible to use dynamic ordering to expand targets that can most improve the GBF score first. However, it was shown that a static ordering can actually make computing upper bounds much faster and ultimately make the search more efficient (Yuan & Hansen, 2009). We therefore simply ordered the targets according to their indices in this work.

3.2 An Upper Bound Based on Target Blanket Decomposition

For MRE inference, an upper bound of a state S should be greater than the GBF score of all descendant states of S. During search, we can keep track of the highest-scoring state and prune the whole subtree if the upper bound is less than the GBF of the current best solution. In the following we introduce a novel upper bound for MRE inference.

We first define a concept called target blanket decomposition of evidence variables.

Definition 2. *A* target blanket decomposition of evidence variables *is a tuple* $< EvdList, TBList >$ *that satisfies the following properties: 1)* $EvdList$ *is a set of* exclusive *and* exhaustive *subsets, i.e.,* $EvdList = \{\mathbf{E}_i\}$, $\mathbf{E} = \cup_i \mathbf{E}_i$ *and* $\mathbf{E}_i \cap \mathbf{E}_j = \emptyset$ *for* $\forall i \neq j$. *2)* $TBList$ *is a set of target blankets, one blanket* $TB(\mathbf{E}_i)$ *for each evidence subset* $\mathbf{E}_i$*, such that* $TB(\mathbf{E}_i)$ *is the minimal set of targets which d-separates* $\mathbf{E}_i$ *from other targets and evidence variables.*

The target blanket decomposition naturally decomposes the whole network into overlapping subnetworks, with each subnetwork containing an evidence subset $\mathbf{E}_i$ and its *target blanket* $TB(\mathbf{E}_i)$. Target blankets are so named due to their resemblance to Markov blankets, but strictly speaking, they are not really the Markov blankets of the evidence variables, because there may be auxiliary variables between them.

For any given target set $\mathbf{M}$ and evidence set $\mathbf{E}$, there may exist multiple target blanket decompositions that satisfy Definition 2. To see that, note that we can simply merge two or more evidence subsets and their corresponding target blankets in a decomposition to obtain another decomposition. However, among all possible decompositions, there is a minimal target blanket decomposition, which we define as follows.

Definition 3. *A* minimal *target blanket decomposition of evidence variables is a target blanket decomposition* $< EvdList, TBList >$ *in which no proper subset of any* $\mathbf{E}_i$ *in* $EvdList$ *has a valid target blanket.*

The minimal target blanket decomposition must be unique according to the following theorem.

Theorem 1. *The minimal target blanket decomposition of evidence set* $\mathbf{E}$ *is unique.*

Proof. Proof by contradiction: assume there are two minimal target blanket decompositions D_1 and D_2. Based on the property of minimal target blanket decomposition, no evidence subset of D_1 can be a proper subset of any evidence subset of D_2, and vice versa. There must exist two distinct evidence subsets $\mathbf{E}_i^{D_1}$ of D_1 and $\mathbf{E}_j^{D_2}$ of D_2 whose overlap $\mathbf{E}_{ij}$ is not empty. But then it must be true that $\mathbf{E}_{ij}$ is d-separated from all other target and evidence variables given some subset of $TB(\mathbf{E}_i^{D_1}) \cup TB(\mathbf{E}_j^{D_2})$, contradicting the definition. □

It immediately follows that all non-minimal target blanket decompositions can be generated by performing merging on the minimal decomposition.

An upper bound on GBF is then derived by multiplying the upper bounds on belief update ratios calculated on the subnetworks. We first derive an upper bound on the belief update ratio in the following theorem.

Theorem 2. *Let* $\mathbf{M} = \{X_1, X_2, \ldots, X_n\}$ *be the set of targets,* $\mathbf{e}$ *be the evidence, and* $< EvdList$, $TBList >$ *be a target blanket decomposition of* $\mathbf{E}$*, where* $EvdList = \{\mathbf{E}_i\}$ *and* $TBList = \{TB(\mathbf{E}_i)\}$*. Then, for any subset* $\emptyset \subset \mathbf{X} \subseteq \mathbf{M}$*, the belief update ratio* $r(\mathbf{x}; \mathbf{e})$ *is upper bounded as follows:*

$$\max_{\mathbf{x}, \emptyset \subset \mathbf{X} \subseteq \mathbf{M}} r(\mathbf{x}; \mathbf{e}) \leq \left(\prod_i \max_{\mathbf{z}, \mathbf{Z} = TB(\mathbf{E}_i)} r(\mathbf{z}; \mathbf{e}_i) \right) \cdot C, \tag{6}$$

where $C = \prod_i p(\mathbf{e}_i)/p(\mathbf{e})$*.*

Proof. From the formulation of $r(\mathbf{M};\mathbf{e})$, we have

$$\begin{aligned} r(\mathbf{M};\mathbf{e}) &= p(\mathbf{M}|\mathbf{e})/p(\mathbf{M}) \\ &= p(\mathbf{e}|\mathbf{M})/p(\mathbf{e}) \\ &= \prod_i p(\mathbf{e}_i|\,TB(\mathbf{E}_i))/p(\mathbf{e}) \\ &= \prod_i \frac{p(\,TB(\mathbf{E}_i)|\mathbf{e}_i)p(\mathbf{e}_i)}{p(\,TB(\mathbf{E}_i))}/p(\mathbf{e}) \\ &= \left(\prod_i r(\,TB(\mathbf{E}_i);\mathbf{e}_i)p(\mathbf{e}_i)\right)/p(\mathbf{e}). \end{aligned} \tag{7}$$

The third equality is based on the property of target blankets. Thus, we have

$$r(\mathbf{M};\mathbf{e}) = \left(\prod_i r(\,TB(\mathbf{E}_i);\mathbf{e}_i)\right)\cdot C, \tag{8}$$

where $C = \prod_i p(\mathbf{e}_i)/p(\mathbf{e})$. From Equation 8, we immediately have the following upper bound on $r(\mathbf{m};\mathbf{e})$,

$$\max_{\mathbf{m}} r(\mathbf{m};\mathbf{e}) \le \left(\prod_i \max_{\mathbf{z},\mathbf{Z}=TB(\mathbf{E}_i)} r(\mathbf{z};\mathbf{e}_i)\right)\cdot C. \tag{9}$$

For any $\emptyset \subset \mathbf{X} \subset \mathbf{M}$, $\bar{\mathbf{X}} = \mathbf{M}\backslash\mathbf{X}$, let $\mathbf{S}_i = TB(\mathbf{E}_i) \cap \bar{\mathbf{X}}$ denote the subset of targets to be summed out in the i^{th} subnetwork, and $\bar{\mathbf{S}}_i = TB(\mathbf{E}_i) \cap \mathbf{X}$. Thus, we separate $TB(\mathbf{E}_i)$ into two parts, i.e., related or unrelated to the summation on $\bar{\mathbf{X}}$, indexed by $\mathbf{I} = \{i : \mathbf{S}_i \neq \emptyset\}$ and $\mathbf{J} = \{j : \mathbf{S}_j = \emptyset\}$ respectively. Based on the above definition, we have:

$$\begin{aligned} p(\mathbf{e}|\mathbf{X}) &= \sum_{\bar{\mathbf{X}}} p(\mathbf{e}|\mathbf{M})p(\bar{\mathbf{X}}|\mathbf{X}) \\ &= \sum_{\bar{\mathbf{X}}} \left(\prod_{i\in\mathbf{I}} p(\mathbf{e}_i|\,TB(\mathbf{E}_i))\right) p(\bar{\mathbf{X}}|\mathbf{X}) \cdot \prod_{j\in\mathbf{J}} p(\mathbf{e}_j|\,TB(\mathbf{E}_j)) \\ &\le \left(\prod_{i\in\mathbf{I}} \max_{\mathbf{s}_i} p(\mathbf{e}_i|\bar{\mathbf{S}}_i \cup \mathbf{s}_i)\right)\cdot\left(\prod_{j\in\mathbf{J}} p(\mathbf{e}_j|\,TB(\mathbf{E}_j))\right). \end{aligned} \tag{10}$$

The last inequality is derived by using maximization instead of averaging on $\bar{\mathbf{X}}$. Since $p(\mathbf{e}|\mathbf{X}) = r(\mathbf{X};\mathbf{e})p(\mathbf{e})$, we have the following upper bound on $r(\mathbf{x};\mathbf{e})$ by performing maximization according to $\mathbf{X}$ on both sides of Equation 10:

$$\begin{aligned} \max_{\mathbf{x}} r(\mathbf{x};\mathbf{e}) &= \max_{\mathbf{x}} \frac{p(\mathbf{e}|\mathbf{x})}{p(\mathbf{e})} \\ &\le \left(\prod_{i\in\mathbf{I}} \max_{\bar{\mathbf{s}}_i}\max_{\mathbf{s}_i} \frac{p(\mathbf{e}_i|\mathbf{s}_i\cup\bar{\mathbf{s}}_i)p(\mathbf{e}_i)}{p(\mathbf{e}_i)}\right)\cdot\left(\prod_{j\in\mathbf{J}} \max_{\mathbf{z},\mathbf{Z}=TB(\mathbf{E}_j)} \frac{p(\mathbf{e}_j|\mathbf{z})p(\mathbf{e}_j)}{p(\mathbf{e}_j)}\right)\cdot\frac{1}{p(\mathbf{e})} \\ &= \left(\prod_{i\in\mathbf{I}} \max_{\mathbf{z},\mathbf{Z}=TB(\mathbf{E}_i)} r(\mathbf{z};\mathbf{e}_i)\right)\cdot\left(\prod_{j\in\mathbf{J}} \max_{\mathbf{z},\mathbf{Z}=TB(\mathbf{E}_j)} r(\mathbf{z};\mathbf{e}_j)\right)\cdot C. \end{aligned} \tag{11}$$

Thus, for any $\emptyset \subset \mathbf{X} \subseteq \mathbf{M}$, we obtain the final upper bound by combining Equations 9 and 11.

$$\max_{\mathbf{x}, \emptyset \subset \mathbf{X} \subseteq \mathbf{M}} r(\mathbf{x}; \mathbf{e}) \leq \left(\prod_i \max_{\mathbf{z}, \mathbf{Z} = TB(\mathbf{E}_i)} r(\mathbf{z}; \mathbf{e}_i) \right) \cdot C.$$

□

In MRE inference, the evidence $\mathbf{e}$ is given, thus C is a constant. Theorem 2 assumes that $\mathbf{V} = \emptyset$, which is true at the beginning of search. During the search when $\mathbf{V} \neq \emptyset$, we have the following corollary to derive the upper bound on the belief update ratio.

Corollary 3. *Let* $\mathbf{M} = \{X_1, X_2, \ldots, X_n\}$ *be the set of targets,* $\mathbf{e}$ *be the evidence, and* $< EvdList, TBList >$ *be a target blanket decomposition of* $\mathbf{E}$*, where* $EvdList = \{\mathbf{E}_i\}$ *and* $TBList = \{TB(\mathbf{E}_i)\}$*. Let* $\mathbf{U}$ *and* $\mathbf{V}$ *be the unexpanded and expanded target sets of state* S*. Let* $\mathbf{T}_i = TB(\mathbf{E}_i) \cap \mathbf{V}$*, and* $\bar{\mathbf{T}}_i = TB(\mathbf{E}_i) \backslash \mathbf{T}_i$*. Then, for any subset* $\emptyset \subset \mathbf{X} \subseteq \mathbf{U}$*, the belief update ratio* $r(\mathbf{x} \cup \mathbf{v}; \mathbf{e})$ *is upper bounded as follows.*

$$\max_{\mathbf{x}, \emptyset \subset \mathbf{X} \subseteq \mathbf{U}} r(\mathbf{x} \cup \mathbf{v}; \mathbf{e}) \leq \left(\prod_i \max_{\mathbf{z}, \mathbf{Z} = \bar{\mathbf{T}}_i} r(\mathbf{z} \cup \mathbf{t}_i; \mathbf{e}_i) \right) \cdot C, \tag{12}$$

where $C = \prod_i p(\mathbf{e}_i)/p(\mathbf{e})$.

Based on Corollary 3, we can derive an upper bound on GBF in Theorem 4:

Theorem 4. *Let* $\mathbf{M} = \{X_1, X_2, \ldots, X_n\}$ *be the set of targets,* $\mathbf{e}$ *be the evidence, and* $< EvdList, TBList >$ *be a target blanket decomposition of* $\mathbf{E}$*, where* $EvdList = \{\mathbf{E}_i\}$ *and* $TBList = \{TB(\mathbf{E}_i)\}$*. Let* $\mathbf{U}$ *and* $\mathbf{V}$ *be the unexpanded and expanded target sets of state* S*. Let* $\mathbf{T}_i = TB(\mathbf{E}_i) \cap \mathbf{V}$*, and* $\bar{\mathbf{T}}_i = TB(\mathbf{E}_i) \backslash \mathbf{T}_i$*. Then, for any subset* $\emptyset \subset \mathbf{X} \subseteq \mathbf{U}$*, the generalized Bayesian factor score* $GBF(\mathbf{x} \cup \mathbf{v}; \mathbf{e})$ *is upper bounded as follows.*

$$\max_{\mathbf{x}, \emptyset \subset \mathbf{X} \subseteq \mathbf{U}} GBF(\mathbf{x} \cup \mathbf{v}; \mathbf{e}) \leq 1 + \frac{\prod_i \max_{\mathbf{z}, \mathbf{Z} = \bar{\mathbf{T}}_i} r(\mathbf{z} \cup \mathbf{t}_i; \mathbf{e}_i) \cdot C - 1}{1 - p(\mathbf{v}|\mathbf{e})}, \tag{13}$$

where $C = \prod_i p(\mathbf{e}_i)/p(\mathbf{e})$.

Proof. First, we formulate GBF using the belief update ratio as in Equation 5.

$$GBF(\mathbf{m}; \mathbf{e}) = 1 + \frac{r(\mathbf{m}; \mathbf{e}) - 1}{1 - p(\mathbf{m}|\mathbf{e})}.$$

For any subset $\emptyset \subset \mathbf{X} \subseteq \mathbf{U}$, $p(\mathbf{x} \cup \mathbf{v}|\mathbf{e}) = p(\mathbf{x}|\mathbf{v}, \mathbf{e}) \cdot p(\mathbf{v}|\mathbf{e}) \leq p(\mathbf{v}|\mathbf{e})$. Thus we have

$$\max_{\mathbf{x}, \emptyset \subset \mathbf{X} \subseteq \mathbf{U}} GBF(\mathbf{x} \cup \mathbf{v}; \mathbf{e}) \leq 1 + \frac{\max_{\mathbf{x}, \emptyset \subset \mathbf{X} \subseteq \mathbf{U}} r(\mathbf{x} \cup \mathbf{v}; \mathbf{e}) - 1}{1 - p(\mathbf{v}|\mathbf{e})}. \tag{14}$$

Then using Corollary 3, we obtain the following upper bound on GBF:

$$\max_{\mathbf{x}, \emptyset \subset \mathbf{X} \subseteq \mathbf{U}} GBF(\mathbf{x} \cup \mathbf{v}; \mathbf{e}) \leq 1 + \frac{\left(\prod_i \max_{\mathbf{z}, \mathbf{Z} = \bar{\mathbf{T}}_i} r(\mathbf{z} \cup \mathbf{t}_i; \mathbf{e}_i) \right) \cdot C - 1}{1 - p(\mathbf{v}|\mathbf{e})},$$

where $C = \prod_i p(\mathbf{e}_i)/p(\mathbf{e})$.

□

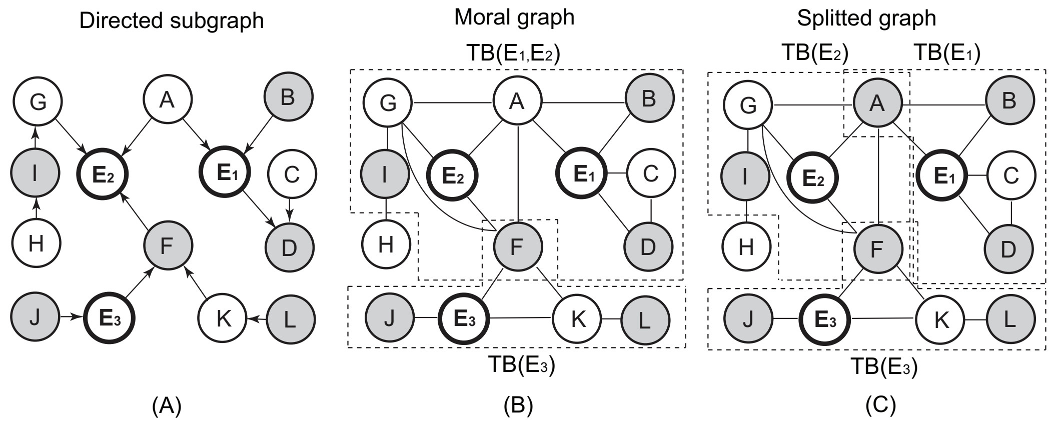

Figure 3: An example of compiling and splitting target blanket decomposition. E_1, E_2, and E_3 are evidence nodes. Gray nodes are targets. Others are auxiliary nodes. The original target blanket $TB(E_1, E_2)$ is split into $TB(E_1)$ and $TB(E_2)$ by converting auxiliary node A into pseudo-target.

Using Equation 13, we can bound all the descendant states of the current state S. Equation 13 shows that in an MRE problem (left), we need to search all of the subsets of targets $\mathbf{M}$ to find the best solution. However, to calculate an upper bound (right), we only need to search a fixed target set $TB(\mathbf{E}_i)$ of each subnetwork, which usually has a small size and is easy to compute. For each instantiation $\mathbf{z}$ of $TB(\mathbf{E}_i)$, we calculate $r(\mathbf{z}; \mathbf{e}_i)$ and store it in a table called *belief ratio table*. There is one such table for each subnetwork. These tables are computed in a preprocessing step and are looked up at each step of the search to compute the upper bound of a state.

3.3 Compiling Minimal Target Blanket Decomposition

Theorems 2- 4 are based on factorizing the conditional joint distribution $p(\mathbf{e}|\mathbf{M})$ into the product of a set of conditional distributions $\prod_i p(\mathbf{e}_i | TB(\mathbf{E}_i))$. Thus finding the target blanket decomposition, including the evidence subsets and their target blankets, is a key part of the proposed methods.

In the proposed method, we compile the minimal target blanket decomposition based on Lemma 1 by first compiling a moral graph. The moral graph is used to prune out irrelevant parts of the network and set up the network for separation tests. To compile moral graph, we first generate the smallest ancestral set containing the target set $\mathbf{M}$ and the evidence set $\mathbf{E}$, i.e., $AN(\mathbf{M} \cup \mathbf{E})$. Then we compile the moral graph $(G_{AN(\mathbf{M} \cup \mathbf{E})})^m$. Figure 3(B) illustrates an example moral graph compiled from the Bayesian network with three evidence nodes in Figure 3(A). Using Lemma 1, minimal target blanket decomposition of evidence variables is achieved by doing a depth first graph traversal starting from each unvisited evidence node in the moral graph $(G_{AN(\mathbf{M} \cup \mathbf{E})})^m$. There are three scenarios when a node is being visited.

- **Case 1:** When an evidence node is visited, we add the evidence to the current evidence subset $\mathbf{E}_i$, mark it as visited, and continue to visit its unmarked neighbors.

- **Case 2:** When a target is visited, we add the target to $TB(\mathbf{E}_i)$ and mark it as visited.

Algorithm 1 Compiling Minimal Target Blanket Decomposition

Input: $\mathbf{M}$ – target set; $\mathbf{E}$ – evidence set; $MGraph$ – moral graph $(G_{AN(\mathbf{M}\cup\mathbf{E})})^m$.

Output: Target blanket decomposition $< EvdList, TBList >$, where $EvdList = \{\mathbf{E}_i\}$, and $TBList = \{TB(\mathbf{E}_i)\}$.

```
function CompileMinimalTargetBlankets(M, E, MGraph)
    EvdList ← ∅;
    TBList ← ∅;
    for each node Nd in MGraph do
        if Nd in E and Nd is not visited then
            EvdSet_i ← ∅;
            TBSet_i ← ∅;
            SearchStack ← Nd;                ▷ initialize the stack for Depth-first search
            while SearchStack is not empty do
                SearchNode ← SearchStack.pop();
                if SearchNode is visited then
                    continue;
                end if
                mark SearchNode as visited;
                if SearchNode in E then                                    ▷ Case 1
                    EvdSet_i.push(SearchNode);
                    SearchStack.push(SearchNode's unvisited neighbors);
                else if SearchNode in M then                               ▷ Case 2
                    TBSet_i.push(SearchNode);
                else if SearchNode is auxiliary node then                  ▷ Case 3
                    SearchStack.push(SearchNode's unvisited neighbors);
                end if
            end while
            EvdList.push(EvdSet_i);
            TBList.push(TBSet_i);
        end if
        mark the targets and auxiliary nodes in MGraph as unvisited;
    end for
end function
```

- **Case 3:** When an auxiliary node is visited, we mark it as visited and continue to visit its unmarked neighbors.

When restarting the search on a new evidence node, we unmark all targets and auxiliary nodes, because the same targets may occur in different target blankets, e.g., node F is shared by $TB(E_1, E_2)$ and $TB(E_3)$ as shown in Figure 3(B). The algorithm stops when all evidence nodes have been visited. The above algorithm is summarized in Algorithm 1. Furthermore, we show that Algorithm 1 is guaranteed to find the minimal target blanket decomposition of evidence set $\mathbf{E}$.

Theorem 5. *Algorithm 1 is guaranteed to find the minimal target blanket decomposition of evidence variables.*

Proof. For each evidence E, Algorithm 1 stops a search path if and only if a target is encountered. All evidence variables that belong to the same evidence subset $\mathbf{E}_i$ as E in the minimal target blanket decomposition must be visited before all search paths terminated. Furthermore, since we start the search from one of the evidence E in $\mathbf{E}_i$, it must be true that the search stops once the minimal target blanket $TB(\mathbf{E}_i)$ is fully visited. □

3.4 Merging Target Blankets

When computing the upper bound, we maximize the belief update ratio $r(TB(\mathbf{E}_i); \mathbf{e}_i)$ in each $TB(\mathbf{E}_i)$ independently. Thus the common targets of two different target blankets $TB(\mathbf{E}_i)$ and $TB(\mathbf{E}_j)$ may be set to inconsistent values. Too much inconsistency may result in a loose bound. We can tighten the upper bound by merging target blankets that share targets. On the other hand, if the number of targets in any individual target blanket is too large, it will make calculating the belief update ratio tables inefficient or even infeasible due to excessive memory consumption. We propose to merge target blankets that share targets under the constraint that the number of targets in a resulting target blanket cannot exceed a constant K.

We can use an undirected graph to represent the problem of merging target blankets. The nodes in the graph denote target blankets. If two target blankets share targets, there is an edge between the two corresponding nodes. The weight of the edge is the number of targets shared by the two target blankets. This formulation translates the problem of merging target blankets into a graph partition problem. More specifically, the merging problem can be addressed by recursively solving the minimum bisection problem (Feige & Krauthgamer, 2002) on the undirected graph, which partitions the vertices into two equal halves so as to minimize the sum of weights of the edges between the two partitions.

The minimum bisection problem is an NP-hard problem, however. We cannot afford to spend too much time on computing the upper bound. We therefore use a hierarchical clustering-like greedy algorithm for merging the target blankets. We first merge any pair of target blankets if one of them covers the other. Then for all remaining target blankets, we repeatedly merge two target blankets that share the highest number of targets as long as the number of targets in the resulting target blanket does not exceed K. The algorithm iterates the above two steps until no target blankets can be merged. The merge algorithm is summarized in Algorithm 2.

4. An Improved Target Blanket Bound

The above target blanket upper bound has two potential difficulties in scaling to large Bayesian networks with many target variables. First, the minimal target blanket decomposition (in Section 3.3) can lead to large subnetworks with belief ratio tables that are too large to build. Second, the upper bound can be still loose even after merging target blankets. In this section, we propose another upper bound based on two new techniques for improving the scalability and tightness of the previous bound.

4.1 Splitting Large Target Blankets

The decomposition method in Section 3.3 may lead to large target blankets which prevent the application of BFBnB to large scale MRE problems. To address this problem, we notice that in MRE problems, each subnetwork contains three types of nodes, i.e., targets $TB(\mathbf{E}_i)$, evidence $\mathbf{E}_i$, and

Algorithm 2 Merging Target Blankets

Input: A target blanket decomposition $< EvdList, TBList >$, where $EvdList = \{\mathbf{E}_i\}$, and $TBList = \{TB(\mathbf{E}_i)\}$; K – the maximum number of targets in a target blanket.

Output: An updated decomposition $< EvdList, TBList >$.

function MERGETARGETBLANKETS($EvdList$, $TBList$, K)
 $MergeFlag \leftarrow$ True;
 while $MergeFlag$ **do**
  $TBListOld \leftarrow TBList$;   ▷ TB_i denotes $TB(\mathbf{E}_i)$
  $EvdListOld \leftarrow EvdList$;
  for each (TB_i, TB_j) pair, where $TB_i, TB_j \in TBListOld$ **do**
   if TB_i and TB_j in $TBList$ **then**   ▷ merge target blankets with subset relation
    if $TB_i \subseteq TB_j$ **then**
     TB_j.merge(TB_i);
     $TBList$.remove(TB_i);
     $\mathbf{E}_j$.merge($\mathbf{E}_i$);
     $EvdList$.remove($\mathbf{E}_i$);
    else if $TB_j \subseteq TB_i$ **then**
     TB_i.merge(TB_j);
     $TBList$.remove(TB_j);
     $\mathbf{E}_i$.merge($\mathbf{E}_j$);
     $EvdList$.remove($\mathbf{E}_j$);
    end if
   end if
  end for
  $NumTargets \leftarrow 0$;
  for each (TB_i, TB_j) pair, where $TB_i, TB_j \in TBList$ **do**
   if $(TB_i \cap TB_j)$.size() $> NumTargets$ and $(TB_i \cup TB_j)$.size() $< K$ **then**
    $TBPair \leftarrow (TB_i, TB_j)$;   ▷ find an TB pair to merge
    $NumTargets \leftarrow (TB_i \cap TB_j)$.size();
    $EvdPair \leftarrow (\mathbf{E}_i, \mathbf{E}_j)$;
   end if
  end for
  if $NumTargets$ is 0 **then**
   $MergeFlag \leftarrow$ False;
  else
   $TBPair.TB_i$.merge($TBPair.TB_j$);
   $TBList$.remove($TBPair.TB_j$);
   $EvdPair.\mathbf{E}_i$.merge($EvdPair.\mathbf{E}_j$);
   $EvdList$.remove($EvdPair.\mathbf{E}_j$);
  end if
 end while
end function

auxiliary nodes $\mathbf{A}_i$, where $TB(\mathbf{E}_i)$ marks the boundary of subnetworks, and $\mathbf{E}_i$ and $\mathbf{A}_i$ are conditionally independent from outside nodes given $TB(\mathbf{E}_i)$. Our idea here is to split large target blankets by converting auxiliary nodes into pseudo-targets which only act as targets in compiling target blanket decomposition and calculating belief ratio tables. These pseudo-targets add additional d-separation to the Bayesian network, thus split the large target blankets into smaller ones. Theorems 4 and 6 guarantee that, after the split, we can still find an upper bound on GBF.

Theorem 6. *Let $TB(\mathbf{E}_i)$ be the target blanket of the i^{th} subset of evidence $\mathbf{E}_i$, and $\mathbf{A}_i$ be the set of auxiliary nodes in the i^{th} subnetwork of a target blanket decomposition. Assuming that after converting any subset $\emptyset \subset \mathbf{Y} \subseteq \mathbf{A}_i$ into pseudo-targets, the i^{th} subnetwork is decomposed into a set of target blankets $TB(\mathbf{E}_{ij})$, where $TB(\mathbf{E}_i) \cup \mathbf{Y}$=$\cup_j TB(\mathbf{E}_{ij})$, $\mathbf{E}_i$=$\cup_j \mathbf{E}_{ij}$, and $\mathbf{E}_{ij} \cap \mathbf{E}_{ik} = \emptyset$ for $\forall j \neq k$. Then, the belief update ratio $r(\mathbf{x}; \mathbf{e}_i)$ is upper bounded as follows.*

$$\max_{\mathbf{x}, \mathbf{X}=TB(\mathbf{E}_i)} r(\mathbf{x}; \mathbf{e}_i) \leq \prod_j \max_{\mathbf{z}, \mathbf{Z}=TB(\mathbf{E}_{ij})} r(\mathbf{z}; \mathbf{e}_{ij}) \cdot C, \tag{15}$$

where $C = \prod_j p(\mathbf{e}_{ij}) / p(\mathbf{e}_i)$.

Proof. Let $\mathbf{X} = TB(\mathbf{E}_i)$, for any $\emptyset \subset \mathbf{Y} \subseteq \mathbf{A}_i$ we have,

$$\begin{aligned} p(\mathbf{e}_i|\mathbf{X}) &= \sum_{\mathbf{Y}} p(\mathbf{e}_i|\mathbf{X} \cup \mathbf{Y}) p(\mathbf{Y}|\mathbf{X}) \\ &= \sum_{\mathbf{Y}} \prod_j p(\mathbf{e}_{ij}|TB(\mathbf{E}_{ij})) p(\mathbf{Y}|\mathbf{X}) \\ &\leq \prod_j \max_{\mathbf{z}, \mathbf{Z}=TB(\mathbf{E}_{ij}) \cap \mathbf{Y}} p(\mathbf{e}_{ij}|TB(\mathbf{E}_{ij}) \backslash \mathbf{Z}, \mathbf{z}). \end{aligned} \tag{16}$$

The second equality is based on the property of target blankets.

Since $p(\mathbf{e}_i|\mathbf{X}) = r(\mathbf{X}; \mathbf{e}_i) p(\mathbf{e}_i)$, we have the following upper bound on $r(\mathbf{x}; \mathbf{e}_i)$,

$$\max_{\mathbf{x}, \mathbf{X}=TB(\mathbf{E}_i)} r(\mathbf{x}; \mathbf{e}_i) \leq \prod_j \max_{\mathbf{z}, \mathbf{Z}=TB(\mathbf{E}_{ij})} r(\mathbf{z}; \mathbf{e}_{ij}) \cdot C,$$

where $C = \prod_j p(\mathbf{e}_{ij}) / p(\mathbf{e}_i)$.

□

We introduce a greedy algorithm to convert auxiliary nodes into pseudo-targets incrementally until the size of each resulting target blanket does not exceed K. In the algorithm, we split the target blankets whose sizes exceed K using the following steps.

- **Step 1:** Calculate the degree (i.e., the number of neighboring nodes) of each auxiliary node A_{ij} in the moral graph, and sort them according to their degrees in descending order.
- **Step 2:** Convert one auxiliary node into pseudo-target according to its order, and perform depth-first search in Section 3.3 to compile minimal target blankets decomposition on the subnetwork.
- **Step 3:** Stop if the size of each resulting target blanket does not exceed K, or repeat Step 2 until all the auxiliary nodes have been converted.

Algorithm 3 Splitting Target Blankets

Input: A target blanket decomposition $< EvdList, TBList >$, where $EvdList = \{\mathbf{E}_i\}$, and $TBList = \{TB(\mathbf{E}_i)\}$; K – the maximum number of targets in a target blanket. $MGraph$ – moral graph $(G_{AN(\mathbf{M} \cup \mathbf{E})})^m$.

Output: An updated decomposition $< EvdList, TBList >$.

function SPLITTARGETBLANKETS($EvdList$, $TBList$, K, $MGraph$)
    for each TB_i in $TBList$ **do**
        if TB_i.size() $> K$ **then**
            for each A_{ij} in $\mathbf{A}_i$ with descending degree **do** $\triangleright$ $\mathbf{A}_i$ is the auxiliary node set in TB_i
                convert A_{ij} to pseudo-target;
                $< EvdList_i, TBList_i > \leftarrow CompileMinimalTargetBlankets(TB_i, \mathbf{E}_i, MGraph)$;
                if TB_{ij}.size() $< K$, for any TB_{ij} in $TBList_i$ **then**
                    break;
                end if
            end for
        end if
    end for
end function

The above algorithm is summarized in Algorithm 3. An example of splitting target blanket is illustrated in Figure 3(C). The original target blanket $TB(\{E_1, E_2\}) = \{I, F, B, D\}$ is split into $TB(E_1) = \{A, B, D\}$ and $TB(E_2) = \{A, F, I\}$ by converting auxiliary node A into a pseudo-target.

4.2 Tightening the Upper Bound

By combining Theorems 4 and 6, we can see that the splitting process makes the upper bound loose because of the maximizations on pseudo-targets. In order to maintain or even improve the search efficiency, we need to re-tighten the bound. The key idea comes from Theorem 4, in which $\bar{\mathbf{T}}_i$ contains two types of variables, i.e., *pruned-targets* and unexpanded targets. In Equation 13, both pruned-targets and unexpanded targets of state S are maximized to derive the upper bound. Since the pruned-targets do not occur in the subsequently generated states, maximizing pruned-targets makes the upper bound loose. However, directly *summing* the pruned-targets in the subnetworks will change the structure of target blanket decomposition, thus makes the upper bound invalid.

We notice that certain targets can be summed out instead of being maximized over without affecting the decomposition. First, we need to define another type of targets called *enclosed-target* set $\mathbf{H}$ of a target blanket in the following.

Definition 4. *In MRE inference,* enclosed-targets $\mathbf{H}$ *of a target blanket are the targets which are conditionally independent from the variables outside the target blanket given the remaining variables in the target blanket.*

In other words, enclosed-targets are the targets which are blocked (d-separated) from outside by the rest of the target blanket. From Definition 4, we can see that each enclosed-target only occurs in one target blanket. As an example in Figure 3(B), J and L are enclosed-targets in $TB(E_3)$=$\{F, J, L\}$.

The idea is then to sum over the enclosed-targets in individual subnetworks $TB(\mathbf{E}_i)$ in order to produce a tighter upper bound. We have the following theorem.

Theorem 7. *Let $\mathbf{M} = \{X_1, X_2, \ldots, X_n\}$ be the set of targets, $\mathbf{e}$ be the evidence, and $< EvdList, TBList >$ be a target blanket decomposition of $\mathbf{E}$, where $EvdList = \{\mathbf{E}_i\}$ and $TBList = \{TB(\mathbf{E}_i)\}$. Let $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{P}$ be the unexpanded, expanded, and pruned target sets of state S. Let $\mathbf{T}_i = TB(\mathbf{E}_i) \cap \mathbf{V}$, and $\bar{\mathbf{T}}_i = TB(\mathbf{E}_i) \backslash \mathbf{T}_i$. Let $\mathbf{H}_i$ be the enclosed-target set in $TB(\mathbf{E}_i)$ and $\mathbf{H}_{is} = \mathbf{H}_i \cap \mathbf{P}$. Then, for any subset $\emptyset \subset \mathbf{X} \subseteq \mathbf{U}$, the belief update ratio $r(\mathbf{x} \cup \mathbf{v}; \mathbf{e})$ is upper bounded as follows.*

$$\max_{\mathbf{x}, \emptyset \subset \mathbf{X} \subseteq \mathbf{U}} r(\mathbf{x} \cup \mathbf{v}; \mathbf{e}) \leq \prod_i \max_{\mathbf{z}, \mathbf{Z} = \bar{\mathbf{T}}_i \backslash \mathbf{H}_{is}} r(\mathbf{z} \cup \mathbf{t}_i; \mathbf{e}_i) \cdot C, \tag{17}$$

where $C = \prod_i p(\mathbf{e}_i)/p(\mathbf{e})$.

Proof. For any $\emptyset \subset \mathbf{X} \subseteq \mathbf{U}$, let $\mathbf{W} = \mathbf{X} \cup \mathbf{V}$ and $\bar{\mathbf{W}} = \mathbf{M} \backslash \mathbf{W}$. Similar with Theorem 2, let $\mathbf{S}_i = TB(\mathbf{E}_i) \cap \bar{\mathbf{W}}$, $\mathbf{I} = \{i : \mathbf{S}_i \neq \emptyset\}$, and $\mathbf{J} = \{j : \mathbf{S}_j = \emptyset\}$, we have,

$$\begin{aligned} p(\mathbf{e}|\mathbf{W}) &= \sum_{\bar{\mathbf{W}}} p(\mathbf{e}|\mathbf{M}) p(\bar{\mathbf{W}}|\mathbf{W}) \\ &= \sum_{\bar{\mathbf{W}}} \Big(\prod_{i \in \mathbf{I}} p(\mathbf{e}_i | TB(\mathbf{E}_i)) \Big) p(\bar{\mathbf{W}}|\mathbf{W}) \cdot \prod_{j \in \mathbf{J}} p(\mathbf{e}_j | TB(\mathbf{E}_j)). \end{aligned} \tag{18}$$

Equation 18 is based on the property of target blankets.

Assuming $\bar{\mathbf{X}} = \mathbf{U} \backslash \mathbf{X}$, we have $\bar{\mathbf{W}} = \mathbf{P} \cup \bar{\mathbf{X}}$ and $\mathbf{H}_{is} \subseteq \mathbf{S}_i \subseteq \bar{\mathbf{W}}$. Since $\mathbf{H}_{is}$ only occurs in $TB(\mathbf{E}_i)$, we perform summation on non-empty $\mathbf{H}_{is}$ in Equation 18 as follows.

$$\begin{aligned} &\sum_{\mathbf{H}_{is}} p(\mathbf{e}_i | TB(\mathbf{E}_i)) p(\mathbf{H}_{is} | \mathbf{M} \backslash \mathbf{H}_{is}) \\ &= \sum_{\mathbf{H}_{is}} \frac{p(\mathbf{e}_i, \mathbf{H}_{is} | TB(\mathbf{E}_i) \backslash \mathbf{H}_{is})}{p(\mathbf{H}_{is} | TB(\mathbf{E}_i) \backslash \mathbf{H}_{is})} p(\mathbf{H}_{is} | \mathbf{M} \backslash \mathbf{H}_{is}) \\ &= \sum_{\mathbf{H}_{is}} p(\mathbf{e}_i, \mathbf{H}_{is} | TB(\mathbf{E}_i) \backslash \mathbf{H}_{is}) \\ &= p(\mathbf{e}_i | TB(\mathbf{E}_i) \backslash \mathbf{H}_{is}), \end{aligned} \tag{19}$$

where $p(\mathbf{H}_{is} | TB(\mathbf{E}_i) \backslash \mathbf{H}_{is})$ = $p(\mathbf{H}_{is} | \mathbf{M} \backslash \mathbf{H}_{is})$, since $\mathbf{H}_{is}$ is conditionally independent with other targets given $TB(\mathbf{E}_i) \backslash \mathbf{H}_{is}$.

Based on Equation 19, we obtain an upper bound by summing out $\mathbf{H}_{is}$ from Equation 18 and replacing the summation on $\cup_i (\mathbf{S}_i \backslash \mathbf{H}_{is})$ with maximization.

$$p(\mathbf{e}|\mathbf{W}) \leq \prod_{i \in \mathbf{I}} \max_{\mathbf{z}, \mathbf{Z} = \mathbf{S}_i \backslash \mathbf{H}_{is}} p(\mathbf{e}_i | TB(\mathbf{E}_i) \backslash \mathbf{S}_i, \mathbf{z}) \cdot \prod_{j \in \mathbf{J}} p(\mathbf{e}_j | TB(\mathbf{E}_j)). \tag{20}$$

Since $p(\mathbf{e}|\mathbf{W}) = r(\mathbf{W}; \mathbf{e}) p(\mathbf{e})$, we have the following upper bound on $r(\mathbf{x} \cup \mathbf{v}; \mathbf{e})$ by re-organizing the partition of $TB(\mathbf{E}_i)$, i.e., from $\mathbf{S}_i$ based partition to $\mathbf{T}_i$ based partition.

$$\max_{\mathbf{x}, \emptyset \subset \mathbf{X} \subseteq \mathbf{U}} r(\mathbf{x} \cup \mathbf{v}; \mathbf{e}) \leq \prod_{i \in \mathbf{I}} \max_{\mathbf{z}, \mathbf{Z} = \bar{\mathbf{T}}_i \backslash \mathbf{H}_{is}} r(\mathbf{z} \cup \mathbf{t}_i; \mathbf{e}_i) \cdot \prod_{j \in \mathbf{J}} \max_{\mathbf{z}, \mathbf{Z} = \bar{\mathbf{T}}_j} r(\mathbf{z} \cup \mathbf{t}_j; \mathbf{e}_j) \cdot C, \tag{21}$$

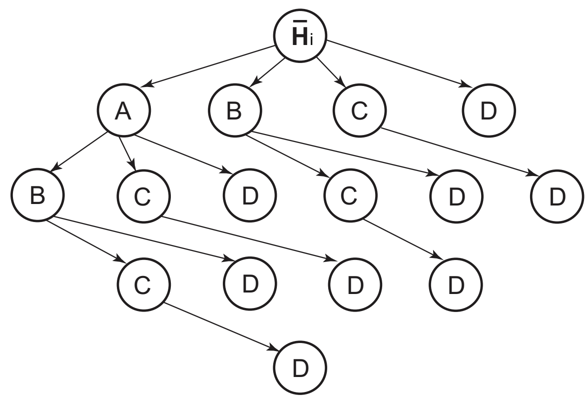

Figure 4: Subproblem graph of compiling belief ratio tables for the incremental algorithm.

where $C = \prod_i p(\mathbf{e}_i)/p(\mathbf{e})$.

Thus, for any $\emptyset \subset \mathbf{X} \subseteq \mathbf{U}$, we obtain the tighter upper bound on $r(\mathbf{x} \cup \mathbf{v}; \mathbf{e})$.

$$\max_{\mathbf{x}, \emptyset \subset \mathbf{X} \subseteq \mathbf{U}} r(\mathbf{x} \cup \mathbf{v}; \mathbf{e}) \leq \prod_i \max_{\mathbf{z}, \mathbf{Z} = \bar{\mathbf{T}}_i \backslash \mathbf{H}_{is}} r(\mathbf{z} \cup \mathbf{t}_i; \mathbf{e}_i) \cdot C.$$

$\square$

By substituting Equation 17 into 14, we have the tightened upper bound on GBF in Theorem 8.

Theorem 8. *Let $\mathbf{M} = \{X_1, X_2, \ldots, X_n\}$ be the set of targets, $\mathbf{e}$ be the evidence, and $< EvdList, TBList >$ be a target blanket decomposition of $\mathbf{E}$, where $EvdList = \{\mathbf{E}_i\}$ and $TBList = \{TB(\mathbf{E}_i)\}$. Let $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{P}$ be the unexpanded, expanded, and pruned target sets of state S. Let $\mathbf{T}_i = TB(\mathbf{E}_i) \cap \mathbf{V}$, and $\bar{\mathbf{T}}_i = TB(\mathbf{E}_i) \backslash \mathbf{T}_i$. Let $\mathbf{H}_i$ be the enclosed-target set in $TB(\mathbf{E}_i)$ and $\mathbf{H}_{is} = \mathbf{H}_i \cap \mathbf{P}$. Then, for any subset $\emptyset \subset \mathbf{X} \subseteq \mathbf{U}$, the generalized Bayesian factor score $GBF(\mathbf{x} \cup \mathbf{v}; \mathbf{e})$ is upper bounded as follows.*

$$\max_{\mathbf{x}, \emptyset \subset \mathbf{X} \subseteq \mathbf{U}} GBF(\mathbf{x} \cup \mathbf{v}; \mathbf{e}) \leq 1 + \frac{\prod_i \max_{\mathbf{z}, \mathbf{Z} = \bar{\mathbf{T}}_i \backslash \mathbf{H}_{is}} r(\mathbf{z} \cup \mathbf{t}_i; \mathbf{e}_i) \cdot C - 1}{1 - p(\mathbf{v}|\mathbf{e})}, \tag{22}$$

where $C = \prod_i p(\mathbf{e}_i)/p(\mathbf{e})$.

The main difference between Theorems 4 and 8 is that Theorem 4 maximizes $\mathbf{H}_{is}$ in the upper bound while Theorem 8 sums out $\mathbf{H}_{is}$ which results in a tighter upper bound.

4.3 Compiling Belief Ratio Tables

The belief ratio tables contain the belief update ratios of all configurations of a series of target sets generated based on $TB(\mathbf{E}_i)$, and are used to calculate upper bounds during MRE inference. To compile belief ratio tables, we find the set of enclosed-targets $\mathbf{H}_i$ by converting each target in $TB(\mathbf{E}_i)$ into an auxiliary node individually. If this does not add new targets or evidence nodes into the original subnetwork, we add the target into $\mathbf{H}_i$. All the subsets of $\mathbf{H}_i$ are used to build $2^{|\mathbf{H}_i|}$ belief ratio tables of $TB(\mathbf{E}_i)$, where $|\mathbf{H}_i|$ denotes the size of $\mathbf{H}_i$. Let $\mathbf{H}_{ij}$ be a subset of $\mathbf{H}_i$, then the target set of the belief ratio table is $TB(\mathbf{E}_i)\backslash\mathbf{H}_{ij}$. Since the number of belief ratio tables increases exponentially according to $|\mathbf{H}_i|$, we limit $|\mathbf{H}_i|$ to be no larger than N for both time and space reasons. Different enclosed-targets contribute different amount in tightening the upper bound, thus we select the top N enclosed-targets from $\mathbf{H}_i$ by sorting the enclosed-targets H_{ij} according to their belief update ratios $max_{h_{ij}} r(h_{ij}; \mathbf{e}_i)$ in descending order.

To compile belief ratio tables, we need to calculate the belief update ratios of all the configurations of target set $TB(\mathbf{E}_i)\backslash\mathbf{H}_{ij}$. A straightforward method is to compile each of the $2^{|\mathbf{H}_i|}$ belief ratio tables independently. But this is slow because of redundant computation. In this work, we propose an incremental algorithm which compiles the belief ratio tables gradually from the smallest target set $\bar{\mathbf{H}}_i$=$TB(\mathbf{E}_i)\backslash\mathbf{H}_i$. In Figure 4, assuming there are four enclosed-targets, i.e., $\mathbf{H}_i = \{A, B, C, D\}$, organized in alphabetical order, and each node in the subproblem graph represents a target. We first compile the belief ratio table of $\bar{\mathbf{H}}_i$, then traverse the subproblem graph in breadth-first order. At each state (node), we compile the belief ratio table by adding the target into the complied belief ratio table of its parent state. After traversing the subproblem graph, the algorithm compiles all belief ratio tables. The above algorithm is summarized in Algorithm 4.

In the proposed algorithm, we calculate a hash key HK_i of each target blanket based on $\mathbf{H}_i$. The belief ratio tables of each target blanket are stored in a hash table and the hash keys are calculated based on the corresponding subset of enclosed-targets $\mathbf{H}_{ij}$.

5. Breadth-First Branch-and-Bound Algorithms

In MRE inference, all of the search nodes are potential solutions except the root node. We can choose from a variety of search methods to explore the search tree, e.g., depth-first search, best-first search, and breadth-first search. Since MRE prunes away independent and less relevant targets, usually the number of targets in the optimal solution is not large. Thus breadth-first search may reach the optimal solutions faster than other search strategies. That is why we choose breadth-first search for MRE inference. The breadth-first search may require more memory to store the unexpanded states in a layer, however.

In order to utilize the proposed upper bounds, our BFBnB algorithms have two major steps: preprocessing and search. The preprocessing step includes compiling minimal target blanket decomposition, merging target blankets sharing targets, splitting large target blankets, and creating the belief ratio tables for each $TB(\mathbf{E}_i)$. A suite of upper bounds can be derived by including different combinations of the above preprocessing modules. Our empirical results show that the target blanket upper bounds in Theorems 4 and 8 show excellent performance.

The search step of BFBnB explores the search tree layer by layer while keeping track of the highest-scoring state, and prunes a state S if its upper bound is less than the current best GBF. The proposed upper bounds lead to two versions of the BFBnB algorithms, MPBnd and SPBnd. MPBnd is based on the upper bound in Theorem 4, which *maximizes* over both the unexpanded targets and *pruned-targets*. Thus, in MPBnd, each target blanket $TB(\mathbf{E}_i)$ has one belief ratio table which is computed by calculating the belief update ratios of all configurations of $TB(\mathbf{E}_i)$. For each state S, we search for a configuration in each belief ratio table that is consistent with the expanded targets $\mathbf{t}_i$ and has the highest belief update ratio. We then calculate the upper bound of S using Theorem 4.

SPBnd is based on the upper bound in Theorem 8, which *sums* out part of the *pruned-targets* based on the structure of Bayesian networks. Thus, in SPBnd, each target blanket $TB(\mathbf{E}_i)$ has a set of belief ratio tables each of which is derived by calculating the belief update ratios of all

Algorithm 4 Compile Belief Ratio Tables

Input: $\mathbf{M}$ – target set; $\mathbf{E}$ – evidence set; $\mathbf{e}$ – given evidence; $MGraph$ – moral graph $(G_{AN(\mathbf{M}\cup\mathbf{E})})^m$; A target blanket decomposition $< EvdList, TBList >$, where $EvdList = \{\mathbf{E}_i\}$, and $TBList = \{TB(\mathbf{E}_i)\}$; N – the maximum number of enclosed-targets in a target blanket.

Output: $BeliefRatioTable$ – the belief ratio tables of each target blanket.

function COMPILEBELIEFRATIOTABLE($\mathbf{M}$, $\mathbf{E}$, $\mathbf{e}$, $MGraph$, $EvdList$, $TBList$, N)
  $BeliefRatioTable \leftarrow \emptyset$;
  for each TB_i in $TBList$ **do**
    $\mathbf{H}_i \leftarrow \emptyset$;
    for each X in TB_i **do** ▷ generate enclosed-target set $\mathbf{H}_i$
      change X to auxiliary node;
      $NewTBList \leftarrow CompileMinimalTargetBlankets(\mathbf{M}, \mathbf{E}, MGraph)$;
      $NewTB_i$.push(X);
      if $NewTB_i$ == TB_i and $NewE_i$ == $\mathbf{E}_i$ **then**
        $\mathbf{H}_i$.push(X);
      end if
      reset X to target node;
    end for
    $\mathbf{H}_i \leftarrow \mathbf{H}_i$.top($N$); ▷ select top N nodes according to $max_{h_{ij}} r(h_{ij}; \mathbf{e}_i)$
    $SearchQueue \leftarrow \emptyset$;
    $BeliefRatioTable_i \leftarrow \emptyset$;
    $\bar{\mathbf{H}}_i = TB_i \backslash \mathbf{H}_i$;
    for each instantiation $\bar{\mathbf{h}}_\mathbf{i}$ of $\bar{\mathbf{H}}_i$ **do**
      $BeliefRatioTable_i[Hash(\emptyset)]$.push($GBF(\bar{\mathbf{h}}_\mathbf{i}; \mathbf{e})$);
    end for
    $SearchQueue$.push($\emptyset$); ▷ contain $\mathbf{H}_{ij}$
    while $SearchQueue$ is not empty **do** ▷ compile belief ratio table of TB_i
      $\mathbf{H}_{ij} \leftarrow SearchQueue$.pop();
      for each $X >_\pi \forall Y$ in $\mathbf{H}_{ij}$, where X in $\mathbf{H}_i$ **do** ▷ π is the order defined in $\mathbf{H}_i$
        for each instantiation $\mathbf{x}$ of $(X \cup \mathbf{H}_{ij} \cup \bar{\mathbf{H}}_i)$ **do** ▷ calculate incrementally
          $BeliefRatioTable_i[Hash(X \cup \mathbf{H}_{ij})]$.push($GBF(\mathbf{x}; \mathbf{e})$);
        end for
        $SearchQueue$.push($\mathbf{H}_{ij} \cup X$);
      end for
    end while
    $BeliefRatioTable$.push($BeliefRatioTable_i$);
  end for
end function

configurations of $TB(\mathbf{E}_i)\backslash\mathbf{H}_{ij}$. The algorithm of calculating the belief ratio tables in SPBnd is summarized in Algorithm 4, which includes the algorithm used in MPBnd as a special case. For each state S, we calculate a hash key SK based on the pruned-targets and use $SK\&HK_i$ to index the belief ratio table of each target blanket. Then, we search for a configuration in each selected

Algorithm 5 BFBnB Algorithm Based on Target Blanket Upper Bounds

Input: $\mathbf{M}$ – target set; $\mathbf{E}$ – evidence set; $\mathbf{e}$ – given evidence; $MGraph$ – moral graph $(G_{AN(\mathbf{M}\cup\mathbf{E})})^m$; K – the maximum number of targets in a target blanket; N – the maximum number of enclosed-targets in a target blanket.

Output: $BestExplanation$ – $\arg\max_{\mathbf{x},\emptyset\subset\mathbf{X}\subseteq\mathbf{M}} GBF(\mathbf{x};\mathbf{e})$.

1: **function** BFBnBSearch($\mathbf{M}$, $\mathbf{E}$, $\mathbf{e}$, $MGraph$, K, N)
2: $\quad$ $< EvdList, TBList > \leftarrow CompileMinimalTargetBlankets(\mathbf{M}, \mathbf{E}, MGraph)$;
3: $\quad$ $MergeTargetBlankets(EvdList, TBList, K)$;
4: $\quad$ $SplitTargetBlankets(EvdList, TBList, K, MGraph)$;
5: $\quad$ **if** TB_i.size() $< K$, for any TB_i in $TBList$ **then**
6: $\quad\quad$ return $None$; $\quad$ ▷ stop the algorithm if the size of a TB_i larger than K
7: $\quad$ **end if**
8: $\quad$ $BeliefRatioTable \leftarrow CompileBeliefRatioTable(\mathbf{M},\mathbf{E},\mathbf{e},MGraph,EvdList,TBList,N)$;
9: $\quad$ $openList \leftarrow$ all state x of each $X \in \mathbf{M}$; $\quad$ ▷ initialize $openList$;
10: $\quad$ $maxGBF \leftarrow openList$.top();
11: $\quad$ **while** $openList$ is not empty **do**
12: $\quad\quad$ $\mathbf{x} \leftarrow openList.pop()$;
13: $\quad\quad$ **for** each state y of $Y \in \mathbf{U}$ **do** $\quad$ ▷ $\mathbf{U}$ is the unexpanded set of current state $\mathbf{x}$
14: $\quad\quad\quad$ $sucState \leftarrow \{y\} \cup \mathbf{x}$;
15: $\quad\quad\quad$ $UpperBound \leftarrow CalcUpperBound(sucState, BeliefRatioTable)$;
16: $\quad\quad\quad$ **if** $UpperBound \leq maxGBF$ **then**
17: $\quad\quad\quad\quad$ continue; $\quad$ ▷ use $UpperBound$ to prune the expanded state $sucState$
18: $\quad\quad\quad$ **end if**
19: $\quad\quad\quad$ **if** $maxGBF < GBF(sucState;\mathbf{e})$ **then**
20: $\quad\quad\quad\quad$ $maxGBF \leftarrow GBF(sucState;\mathbf{e})$;
21: $\quad\quad\quad\quad$ $BestExplanation \leftarrow \{y\} \cup \mathbf{x}$;
22: $\quad\quad\quad$ **end if**
23: $\quad\quad\quad$ $openList.push(sucState)$;
24: $\quad\quad$ **end for**
25: $\quad$ **end while**
26: **end function**
27: **function** CalcUpperBound($sucState$, $BeliefRatioTable$)
28: $\quad$ **for** each $BeliefRatioTable_i$ in $BeliefRatioTable$ **do**
29: $\quad\quad$ $BeliefRatioTable_{\mathbf{H}_{ij}} \leftarrow BeliefRatioTable_i[Hash(\mathbf{P})\&Hash(\mathbf{H}_i)]$;
30: $\quad\quad$ $MaxBeliefRatio_i \leftarrow$ the maximum belief ratio consistent with $\mathbf{v}$ in $sucState$;
31: $\quad$ **end for**
32: $\quad$ $UpperBound \leftarrow 1 + \frac{\prod_i MaxBeliefRatio_i \cdot C - 1}{1-p(\mathbf{v}|\mathbf{e})}$;
33: **end function**

belief ratio table that is consistent with the expanded targets $\mathbf{t}_i$ and has the highest belief update ratio. Finally, we calculate the upper bound of S using Theorem 8.

The BFBnB algorithm is summarized in Algorithm 5. The main difference between MPBnd and SPBnd is on the methods used to generate the belief ratio table, i.e., Line 8 in Algorithm 5. To speed up the search process, we sort each belief ratio table of a target blanket in descending order,

Networks	Nodes	Leaves	States	Arcs
Alarm	37	11	2.84	46
Carpo	61	43	2.23	74
Hepar	70	41	2.31	123
Insurance	27	6	3.30	52
Emdec6h	168	117	2.00	261
CPCS179	179	151	2.29	239

Table 1: Benchmark diagnostic Bayesian networks used to evaluate the proposed algorithms.

i.e., higher belief update ratios are closer to the front of the table. Furthermore, in the search tree of MRE, the expanded targets of a state are guaranteed to be included in its descedant states. Thus in the proposed method, we record the indices of the best belief update ratios, one for each belief ratio table, for each expanded state. To calculate the upper bound of a current state, we only need to search the best belief update ratio from the recorded indices of its parent.

6. Experiments

The proposed algorithms are evaluated on six benchmark diagnostic Bayesian networks listed in Table 1, i.e., Alarm (Ala), Carpo (Car), Hepar (Hep), Insurance (Ins), Emdec6h (Emd), and CPCS179 (Cpc) (Beinlich, Suermondt, Chavez, & Cooper, 1989; Binder, Koller, Russell, & Kanazawa, 1997; Onisko, 2003; Pradhan, Provan, Middleton, & Henrion, 1994). Among them, Alarm, Carpo, Hepar, and Insurance are networks with fewer than 100 nodes. Emdec6h and CPCS179 are larger networks with more than 100 nodes. We listed the number of nodes (Nodes), the number of leaf nodes (Leaves), the average number of node states (States), and the number of arcs (Arcs) of the Bayesian networks in Table 1. The experiments were performed on a 2.67GHz Intel Xeon CPU E7 with 512G RAM running a 3.7.10 Linux kernel.

6.1 Experimental Design

Since the proposed algorithms, MPBnd and SPBnd, are the first nontrivial exact MRE algorithms, we had to use a naive Breadth-First Brute-Force search algorithm (BFBF) as the baseline; basically BFBF is BFBnB with the bound set to be infinity. We also included the results of tabu search to indicate the difficulty of the MRE problems. In MPBnd and SPBnd, we set the maximum number of targets in a target blanket K to be 18. In SPBnd, we set the maximum number of enclosed-targets in a target blanket N to be 7. BFBF search can only solve test cases with fewer than 15 targets. To compare the performance among BFBF, MPBnd, and SPBnd, we perform the experiments on two test settings, one with exactly 12 targets (12-target setting) and the other with around 20 targets (difficult-target setting). In the 12-target setting, we randomly generated five test settings of each network, each setting consisting of all leaf nodes as evidence, 12 of the remaining nodes as targets, and others as auxiliary nodes. Then for each setting, we randomly generated 20 configurations of evidence (test cases) by sampling from the prior distributions of the networks.

In the difficult-target setting, we randomly generated five test settings of each network, each setting consisting of all leaf nodes as evidence, around 20 of the remaining nodes as targets, and others as auxiliary nodes. The number of targets is selected so that the test cases are too challenging

#Cases/Time	Ala	Car	Hep	Ins	Emd	Cpc
BFBF	1.7e3	66.1	270.0	1.6e5	212.0	1.3e4
MPBnd	17.0	3.5	14.6	1.6e3	15.1	815.0
SPBnd	2.6	1.5	9.0	72.7	13.1	377.0
T6400	92	100	93	81	95	90
	17.3	25.4	44.8	26.3	89.3	108.0
T3200	85	100	91	77	95	90
	9.1	13.2	22.6	13.7	45.0	55.1
T1600	79	100	86	74	95	90
	4.7	6.9	11.5	7.0	23.2	28.0
T800	76	98	81	73	95	90
	2.4	3.6	5.9	3.6	11.8	14.1
T400	72	98	81	73	95	90
	1.2	1.8	3.0	1.8	6.1	7.0

Table 2: Comparison of SPBnd, MPBnd, BFBF, and tabu on running time in seconds (sec) as well as the accuracy of tabu searches on Bayesian networks in the 12-target setting.

for BFBF but are still solvable by MPBnd and SPBnd. Then for each setting, we randomly generated 20 configurations of evidence (test cases) by sampling from the prior distributions of the networks.

In tabu search, we set the number of search steps since the last improvement L and the maximum number of search steps M according to different network settings. In the 12-target setting, we set L to be 20 and M to be $\{400, 800, 1600, 3200, 6400\}$. In the difficult-target setting, we set L to be 80 and M to be $\{12800, 25600, 51200\}$. To evaluate search performance, we compared the solutions of tabu search and SPBnd, and counted the number of test cases on which tabu search achieved optimal solutions.

6.2 Evaluation of MPBnd and SPBnd in the 12-Target Setting

In Table 2, we compared the proposed MPBnd and SPBnd algorithms with BFBF and tabu search on the test cases with 12-target setting. For tabu search, we listed the number of test cases solved optimally (top) and running time in seconds (bottom) when using different limits on the number of search steps. The running time of MPBnd and SPBnd includes both preprocessing time and search time. MPBnd, SPBnd and BFBF were able to solve the test cases exactly. MPBnd is shown to be significantly faster than BFBF because of the pruning by the upper bound. The running time of SPBnd is further reduced by using the tightened upper bound. T400 is the fastest algorithm with the worst accuracy. With the increase of M, the running time of tabu search increased significantly. However, in most of the networks, tabu search could not solve all the test cases optimally, even using more running time than MPBnd and SPBnd.

To compare MPBnd, SPBnd and BFBF in detail, we computed the average running time of individual test settings for MPBnd, SPBnd and BFBF. Then we illustrate the logarithmic running time pairs, BFBF vs MPBnd and MPBnd vs SPBnd, as points in Figures 5 and 6 respectively. We also draw the contour lines to mark the difference between the logarithmic running time in Figures 5 and 6. For example, the contour line marked by -3 in Figure 5 contains the points on

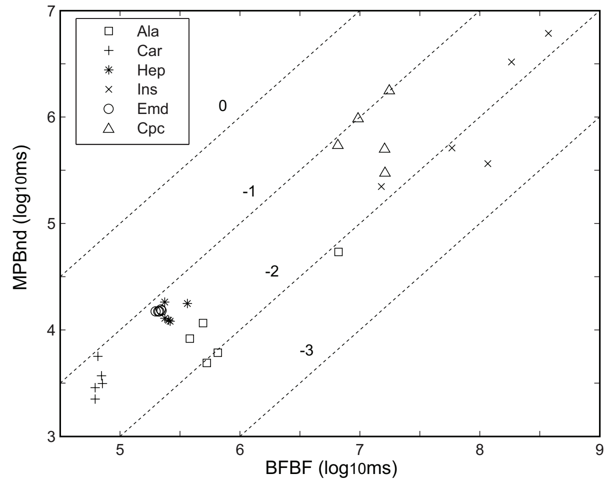

Figure 5: Distributions of logarithmic running time pairs in milliseconds) of MPBnd and BFBF on Bayesian networks in the 12-target setting.

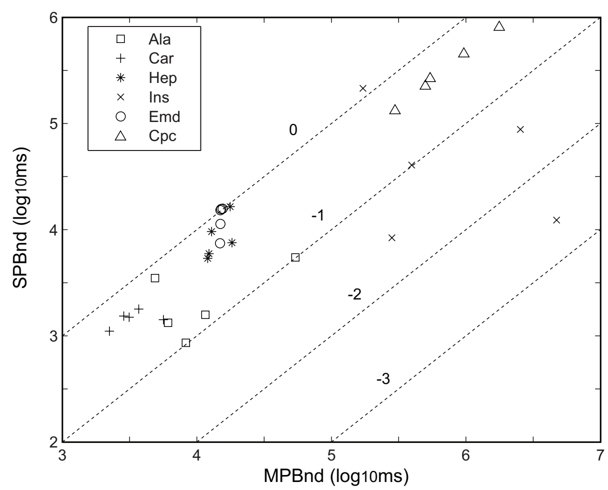

Figure 6: Distributions of logarithmic running time pairs in milliseconds of SPBnd and MPBnd on Bayesian networks in the 12-target setting.

which MPBnd is 1000 times faster than BFBF. The results show that although the average running time may change significantly, the ratios of running times between BFBF and MPBnd, and between MPBnd and SPBnd are relatively stable. MPBnd is roughly 10 to 100 times faster than BFBF. SPBnd is roughly 3 to 4 times faster than MPBnd.

6.3 Evaluation of MPBnd and SPBnd in the Difficult-Target Setting

In Table 3, we compared the proposed SPBnd algorithm with MPBnd and tabu search on the test cases in the difficult-target setting. We list the number of targets of each network in the first row of

#Cases/Time	Ala 20	Car 15	Hep 22	Ins 17	Emd 20	Cpc 20
MPBnd	173	0.98	478	43.91	225	1,569.2
SPBnd	63	0.13	106	58.72	96	419
T51200	60 4.06	100 5.55	72 17.12	76 0.96	86 34.14	82 6.11
T25600	60 2.40	100 2.89	65 9.38	76 0.76	86 17.48	82 4.2
T12800	60 1.42	100 1.50	63 5.15	76 0.59	86 8.93	82 2.7

Table 3: Comparison of SPBnd, MPBnd, and tabu on running time in minutes (min) as well as accuracy of tabu searches on Bayesian networks in the difficult-target setting.

the table. For tabu search, we list both the number of test cases solved optimally (top) and running time in minutes (bottom) on each network. Increasing M from 12800 to 51200 is not helpful in preventing tabu search from getting stuck in the local optima. Moreover, the performance of tabu search varies greatly on different test networks. SPBnd is shown to be significantly faster than MPBnd on most of the networks due to the tightened upper bound. In SPBnd, we need to compile a series of belief ratio tables for each target blanket, which may consume significant amount of running time. For example, although the running time is 419 minutes in CPCS179, the search took only 182.4 minutes. Also from the results, we can see that the running time of MPBnd and SPBnd depends on not only the number of targets, but also the tightness of upper bound and the Bayesian network structures, which control the number of pruned states and the size of belief table in each target blanket $TB(\mathbf{E}_i)$ respectively.

To compare SPBnd and MPBnd in more detail, we computed the average running time of individual test settings for both of them, and illustrated each running time pair in logarithm as a point in Figure 7. We also drew the contour lines to mark the difference between SPBnd and MPBnd. The results showed that the data points of each network form a cluster. SPBnd is again roughly 3 to 4 times faster than MPBnd.

The results in both Figures 6 and 7 showed that for some test cases the running times of MPBnd and SPBnd are close. There are two possible reasons behind the observation. First, for some network structures, the enclosed-target sets may not exist in the individual target blankets. Thus in these cases SPBnd will degenerate to MPBnd. Second, the running time of each test case consists of two parts, compiling belief ratio tables and performing search. For some test cases, compiling belief ratio tables may take significant amount of time in SPBnd. We can therefore make the tradeoff between compiling time and search time by adjusting the maximum number of enclosed-targets N in a target blanket.

6.4 Scalability of SPBnd

We also evaluated MPBnd and SPBnd on test cases with increasing number of targets from 17 to 25 with an increment of 2 on three Bayesian networks, Hepar, Emdec6h, and CPCS179. For each target number i, we randomly generated four test settings and 5 test cases of each setting. In the

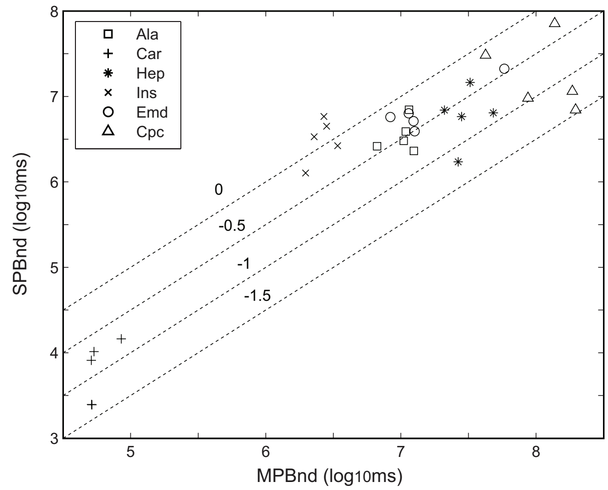

Figure 7: Distributions of logarithmic running time (ms) pairs of SPBnd and MPBnd on test cases in the difficult-target setting.

Time		Targets				
		17	19	21	23	25
MPBnd	Hep	64.7	101.0	411.3	-	-
	Emd	63.6	182.4	643.4	-	-
	Cpc	483.7	-	-	-	-
SPBnd	Hep	24.0	15.0	66.9	243.1	711.9
	Emd	36.3	100.8	444.2	-	-
	Cpc	130.7	94.1	499.6	-	-

Table 4: Comparison of SPBnd and MPBnd on running time (min) on test cases with increasing number of targets. A dash indicates out of time (800m).

Bayesian networks, we set all leaf nodes as evidence, i of the remaining nodes as targets, and others as auxiliary nodes. We set the time limit of running to be 800 minutes. The results are reported in Table 4. The results show that the pruning of upper bound slowed down the exponential growth of running time significantly. SPBnd can handle more complex MRE problems which are out of the reach of MPBnd. From the results, we can also see that the running time of SPBnd is affected by the tightness of upper bound and the structures of Bayesian networks, which no longer heavily depends on the number of targets.

6.5 Importance of Splitting Large Target Blankets

To show the importance of splitting large target blankets, we calculated the percentage of test cases that need splitting operation at various target settings on three Bayesian networks, Hepar, Emdec6h, and CPCS179, with increasing number of targets. In this experiment, we only performed the preprocessing step, so we can handle more than 50 targets. All the target blankets of the test cases in

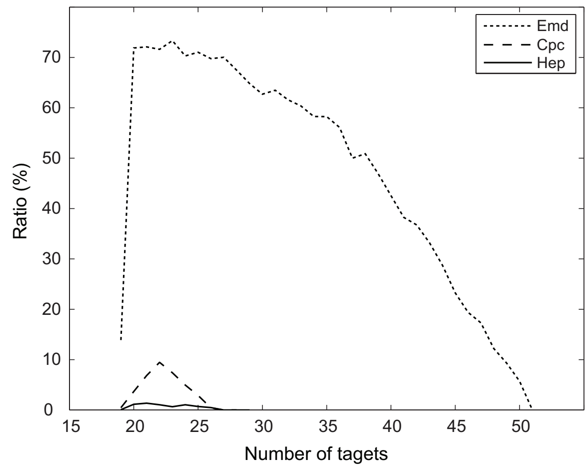

Figure 8: Percentage of cases using splitting operation at different target settings.

Figure 8 can be split into smaller ones with size less than K=18 by using the proposed splitting algorithm. The results in Figure 8 show that the ratio of test cases that needs splitting increases, and then decreases with the increasing numbers of targets. The reason is that initially the increase in the number of targets leads to large target blankets. But as the number of targets keeps increasing, densely distributed targets tend to introduce more d-separation to the Bayesian network and result in smaller target blankets. In Figure 8, the peaks of the curves located at 22 approximately. On the networks with a large number of non-leaf nodes, e.g., Emdec6h, the ratio is higher than 0.5 on a significant number of target settings. This means that without splitting large target blankets most of MRE problems cannot be solved on the networks with these target settings.

6.6 Effect of Summing Out Enclosed-Targets

In this section, we take Hepar, Emdec6h, and CPCS179 as examples to illustrate how summing out enclosed-targets tightens the target blanket upper bound. We calculated the log difference between the upper bound $UpBnd$ and the maximum GBF $MaxGBF$ in each subtree rooted at a search state for the test cases in the 12-target setting. Normalized histograms of the differences for both using and not using the tightening are plotted in Figure 9. The x-axis shows $log_{10}(UpBnd - MaxGBF)$, and the y-axis shows normalized percentages. Thus smaller $log_{10}(UpBnd - MaxGBF)$ indicates tighter upper bound. It is clear from the graph that summing out the enclosed-targets makes the bound much tighter.

6.7 Convergence of Upper Bound

To gain a better perspective on how the tightened upper bound in Theorem 8 improves over time, we calculated both the maximum upper bound $MaxBound$ in each layer of the search tree and the current maximum GBF $CurrMax$. We also recorded the running time when we finished each search layer. Figure 10 shows the convergence curves of $MaxBound$ (dotted line) and $CurrMax$ (circled line) against the search time for one typical test case of each of the three networks, i.e., Hepar, Emdec6h, and CPCS179, under the 17-target setting in Table 4. Figure 10(A) shows that the upper bound dropped sharply at the beginning of the search and decreased gradually during the search.

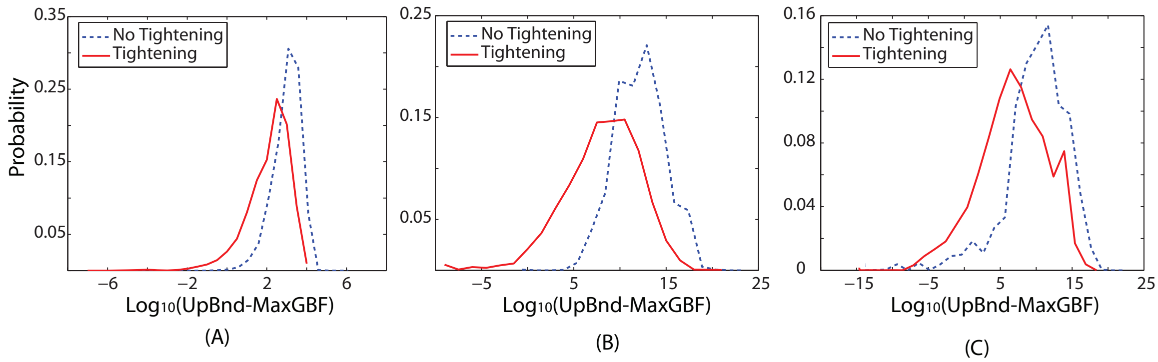

Figure 9: Tightness of upper bounds. We evaluate the effect of summing out enclosed-targets using normalized histogram on different Bayesian networks, i.e., Hepar (A), Emdec6h (B), and CPCS179 (C).

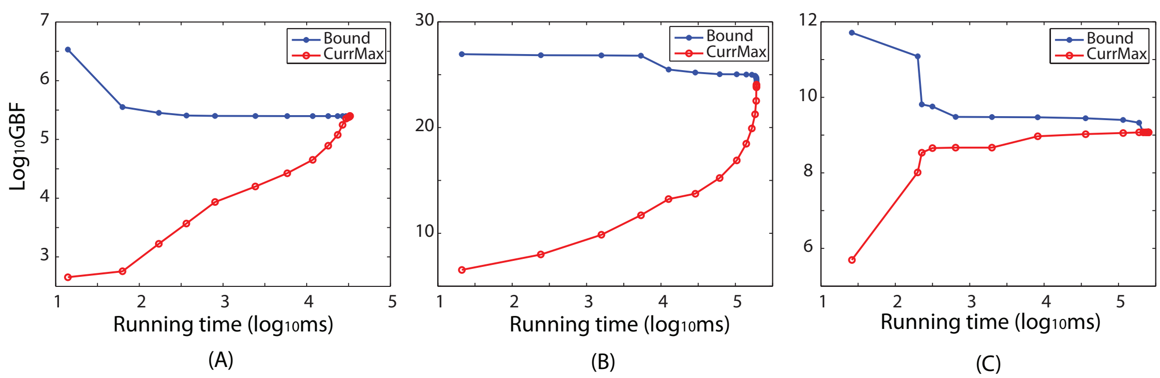

Figure 10: Convergence of upper bound and current maximum GBF on different Bayesian networks, i.e., Hepar (A), Emdec6h (B), and CPCS179 (C).

Figure 10(B) shows that the upper bound dropped sharply at the end of the search and became tight quickly. Figure 10(C) shows that the upper bound dropped sharply at both the beginning and the end of the search. These results demonstrate different behaviors how the bound is tightening during the search.

7. Discussions and Conclusions

The main contributions of this paper are two BFBnB algorithms, i.e., MPBnd and SPBnd, for solving MRE exactly using upper bounds based on target blanket decomposition of evidence variables. These are the first non-trivial exact algorithms for solving MRE in Bayesian networks. The key idea of the proposed method is to decompose the conditional joint probability $p(\mathbf{E}|\mathbf{M})$ into a set of marginal probabilities $p(\mathbf{E}_i|TB(\mathbf{E}_i))$. Each marginal probability is related to a subnetwork characterized by a target blanket of a subset of evidence variables. An upper bound on GBF is derived by maximizing belief update ratio on the fixed target set $TB(\mathbf{E}_i)$ of each subnetwork separately. The upper bound can be tightened by merging the target blankets sharing the same set of targets.

The above bound can be further improved based on two ideas. First, we proposed to split large target blankets by converting auxiliary nodes into pseudo-targets, which scales MRE inference to larger Bayesian networks with more targets. Second, we tightened the upper bound on GBF by identifying and summing out enclosed-targets from each target blanket $TB(\mathbf{E}_i)$. The new upper bound reduces the search space dramatically. The experimental results show that SPBnd is significantly faster than the MPBnd and BFBF algorithms. The proposed SPBnd and MPBnd can solve MRE inference exactly in Bayesian networks which could not be solved previously.

The proposed upper bounds can be calculated efficiently for two reasons. First, each target blanket $TB(\mathbf{E}_i)$ is usually much smaller than the whole target set $\mathbf{M}$. Second, in the original MRE problem, we need to search all the subsets of target set $\mathbf{M}$ to find the best solution. However, to calculate upper bound, we only need to search on a fixed target set $TB(\mathbf{E}_i)$ of each subnetwork.

The proposed MRE upper bound consists of four sources of relaxations, i.e., (1) the relaxation from $p(\mathbf{x}, \mathbf{v}|\mathbf{e})$ to $p(\mathbf{v}|\mathbf{e})$ in Equation 14, (2) bounding all the descendant states, (3) the inconsistent values of the sharing nodes of different target blankets, and (4) the maximization on the pruned-targets of individual search states. The optimal upper bound that can achieved at a state is equal to the maximum GBF in the subtree rooted at the state. When the size of expanded target set increase, both $p(\mathbf{x}, \mathbf{v}|\mathbf{e})$ and $p(\mathbf{v}|\mathbf{e})$ become much smaller than 1, and the relaxation of (1) becomes very tight. Thus, to achieve the optimal upper bound we only need to minimize the impact of (3) and (4). In this work, we minimized the effect of (3) by merging the target blankets sharing the same set of targets, and minimized the effect of (4) by identifying and summing out the enclosed-target sets from each target blanket.

Different from the brute-force algorithm, the search time of MPBnd and SPBnd is no longer mainly dependent on the size of search space (i.e., the number of targets and the number of states of each target), but also on the tightness of the upper bound and the structure of Bayesian networks. For Bayesian networks with a large number of targets, an upper bound can be efficiently generated as long as the the number of targets in each target blanket is small. In SPBnd, the tightness of upper bound depends on the number of enclosed-targets and the quality of each enclosed-target H_{ij} measured by its belief update ratio $max_{h_{ij}} r(h_{ij}; \mathbf{e}_i)$. If there are no enclosed-targets, the proposed method SPBnd degenerates to MPBnd.

In this work, the splitting algorithm is designed to split target blankets, whose sizes are larger than K, based on the separation property of undirected graphs by converting auxiliary nodes into pseudo-targets. If an evidence node is connected directly with many targets, we may decompose it using the methods such as node splitting (Choi et al., 2007). This is also one of our future research directions.

Acknowledgements

This work was supported by NSF grants IIS-0953723, IIS-1219114, and a PSC-CUNY enhancement award. Part of this research has previously been presented in AAAI-15 (Zhu & Yuan, 2015).

References

Ay, N., & Polani, D. (2008). Information flows in causal networks. *Advances in Complex Systems (ACS)*, *11*(01), 17–41.

Beinlich, I., Suermondt, G., Chavez, R., & Cooper, G. (1989). The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the 2nd European Conference on AI and Medicine*, pp. 247–256.

Binder, J., Koller, D., Russell, S., & Kanazawa, K. (1997). Adaptive probabilistic networks with hidden variables. *Machine Learning*, *29*, 213–244.

Chajewska, U., & Halpern, J. Y. (1997). Defining explanation in probabilistic systems. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–97)*, pp. 62–71, San Francisco, CA. Morgan Kaufmann Publishers.

Choi, A., Chavira, M., & Darwiche, A. (2007). Node splitting: A scheme for generating upper bounds in Bayesian networks. In *Proceedings of the 23rd Annual Conference on Uncertainty in Artificial Intelligence (UAI-07)*, pp. 57–66.

Darwiche, A. (2009). *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press.

de Campos, L. M., Gamez, J. A., & Moral, S. (2001). Simplifying explanations in Bayesian belief networks. *International Journal of Uncertainty, Fuzziness Knowledge-Based Systems*, *9*(4), 461–489.

Dechter, R., & Rish, I. (2003). Mini-buckets: A general scheme for bounded inference. *J. ACM*, *50(2)*, 107–153.

Feige, U., & Krauthgamer, R. (2002). A polylogarithmic approximation of the minimum bisection. *SIAM J. Comput.*, *31(4)*, 1090–1118.

Fitelson, B. (2007). Likelihoodism, bayesianism, and relational confirmation. *Synthese*, *156*, 473–489.

Flores, J., Gamez, J. A., & Moral, S. (2005). Abductive inference in Bayesian networks: finding a partition of the explanation space. In *Eighth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, ECSQARU'05*, pp. 63–75. Springer Verlag.

Gärdenfors, P. (1988). *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press.

Glover, F. (1990). Tabu search: A tutorial. *Interfaces*, *20*, 74–94.

Good, I. J. (1985). Weight of evidence: A brief survey. *Bayesian statistics*, *2*, 249–270.

Green, P. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrica*, *82*, 711–732.

Heckerman, D., Breese, J., & Rommelse, K. (1995). Decision-theoretic troubleshooting. *Communications of the ACM*, *38*, 49–57.

Henrion, M., & Druzdzel, M. J. (1991). Qualitative propagation and scenario-based schemes for explaining probabilistic reasoning. In Bonissone, P., Henrion, M., Kanal, L., & Lemmer, J. (Eds.), *Uncertainty in Artificial Intelligence 6*, pp. 17–32. Elsevier Science Publishing Company, Inc., New York, N. Y.

Huang, J., Chavira, M., & Darwiche, A. (2006). Solving map exactly by searching on compiled arithmetic circuits. In *Proceedings of the 21st National Conference on Artificial intelligence*, Vol. 2, pp. 1143–1148. AAAI Press.

Jeffreys, H. (1961). *Theory of Probability*. Oxford University Press.

Jensen, F. V., & Liang, J. (1994). drHugin: A system for value of information in Bayesian networks. In *Proceedings of the 1994 Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pp. 178–183.

Kalagnanam, J., & Henrion, M. (1988). A comparison of decision analysis and expert rules for sequential diagnosis. In *Proceedings of the 4th Annual Conference on Uncertainty in Artificial Intelligence (UAI-88)*, pp. 253–270, New York, NY. Elsevier Science.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, pp. 671–680.

Koller, D., & Friedman, N. (2009). *Probabilistic Graphical Models - Principles and Techniques*. MIT Press.

Kwisthout, J. (2013). Most Inforbable Explanations: Finding explanations in Bayesian networks that are both probable and informative. In van der Gaag, L. (Ed.), *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Vol. 7958 of *Lecture Notes in Computer Science*, pp. 328–339. Springer Berlin Heidelberg.

Lacave, C., & Diez, F. (2002). A review of explanation methods for Bayesian networks. *The Knowledge Engineering Review*, *17*, 107–127.

Lauritzen, S. L., Dawid, A. P., Larsen, B. N., & Leimer., H.-G. (1990). Independence properties of directed markov fields. *Networks*, *20(5)*, 491–505.

Leake, D. B. (1995). Abduction, experience, and goals: A model of everyday abductive explanation. *The Journal of Experimental and Theoretical Artificial Intelligence*, *7*, 407–428.

Marinescu, R., Dechter, R., & Ihler, A. (2014). AND/OR search for marginal MAP. In *Proceedings of the 30th Annual Conference on Uncertainty in Artificial Intelligence (UAI-14)*, pp. 563–572.

Marinescu, R., & Dechter, R. (2009). AND/OR branch-and-bound search for combinatorial optimization in graphical models. *Artif. Intell.*, *173(16-17)*, 1457–1491.

Onisko, A. (2003). *Probabilistic Causal Models in Medicine: Application to Diagnosis of Liver Disorders*. Ph.D. thesis, Institute of Biocybernetics and Biomedical Engineering, Polish Academy of Science.

Pacer, M., Lombrozo, T., Griffiths, T., Williams, J., & Chen, X. (2013). Evaluating computational models of explanation using human judgments. In *Proceedings of the 29th Annual Conference on Uncertainty in Artificial Intelligence (UAI-13)*, pp. 498–507.

Park, J. D., & Darwiche, A. (2003). Solving map exactly using systematic search. In *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)*, pp. 459–468.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.

Pradhan, M., Provan, G., Middleton, B., & Henrion, M. (1994). Knowledge engineering for large belief networks. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence, UAI-94*, p. 484490. Morgan Kaufmann Publishers, Inc.

Shimony, S. (1993). The role of relevance in explanation I: Irrelevance as statistical independence. *International Journal of Approximate Reasoning*, *8*(4), 281–324.

van der Gaag, L., & Wessels, M. (1993). Selective evidence gathering for diagnostic belief networks. *AISB Quarterly*, pp. 23–34.

van der Gaag, L., & Wessels, M. (1995). *Efficient multiple-disorder diagnosis by strategic focusing*, pp. 187–204. UCL Press, London.

Yuan, C., & Hansen, E. A. (2009). Efficient computation of jointree bounds for systematic MAP search. In *Proceedings of 21st International Joint Conference on Artificial Intelligence (IJCAI-09)*, pp. 1982–1989.

Yuan, C., Lim, H., & Littman, M. L. (2011a). Most relevant explanation: computational complexity and approximation methods. *Ann. Math. Artif. Intell.*, *61*, 159–183.

Yuan, C., Lim, H., & Lu, T.-C. (2011b). Most relevant explanation in Bayesian networks. *J. Artif. Intell. Res.*, *42*, 309–352.

Yuan, C., Liu, X., Lu, T.-C., & Lim, H. (2009). Most relevant explanation: Properties, algorithms, and evaluations. In *Proceedings of 25th Annual Conference on Uncertainty in Artificial Intelligence (UAI-09)*, pp. 631–638.

Yuan, C., & Lu, T.-C. (2007). Finding explanations in Bayesian networks. In *Proceedings of the 18th International Workshop on Principles of Diagnosis (DX-07)*, pp. 414–419.

Zhu, X., & Yuan, C. (2015). An exact algorithm for solving most relevant explanation in Bayesian networks. In *Proceedings of the 29th National Conference on Artificial intelligence (AAAI-15)*, pp. 3649–3655.